Human Detection and Face Recognition in Indoor Environment to Improve Human-Robot Interaction in Assistive and Collaborative Robots

by

Sumesh Kunwar

A thesis submitted in Partial fulfillment

of the requirement for the degree of

Masters of Applied Science (M.A.Sc.) in

Natural Resource Engineering

The Faculty of Graduate Studies

Laurentian University,

Sudbury, Ontario, Canada

© Sumesh Kunwar, 2016

Thesis Defense Committee

Abstract

Human detection in indoor environment is essential for Robots working together with humans in collaborative manufacturing environment. Similarly, Human detection is essential for service robots providing service with household chores or helping elderly population with different daily activities.

Human detection can be achieved by Human Head detection, as head is the most discriminative part of human. Head detection method can be divided into three types: ***i***) Method based on color mode; ***ii***) Method based on template matching; and ***iii***) Method based on contour detection. Method based on color mode is simple but is error prone. Method based on head template detects head in the image by searching for a template which is similar to head template. On the other hand, Method based on contour detection uses some information to describe head or head and shoulder information. The use of only one criteria may not be sufficient and accuracy of human head detection can be increased by combining the shape and color information. In this thesis, a method of human detection is proposed by combining the head shape and skin color (i.e., Combination of method based on Color mode and method based on Contour detection). Mainly, curvature criteria is used to segment out curves having similar curvature to find human head. Further, skin color is detected to localize face in image plane. A curve represents human head curve if only it has sufficient skin colored pixel in its closed proximity. Thus, by using color and human head curvature it was found that promising results could be obtained in human detection in indoor environment.

After detecting humans in the surrounding, the next step for the robot could be to identify and recognize them. In this thesis, the use of Gabor filter response on nine points was investigated to identify eight different individuals. This suggests that the Gabor filter on nine points could be applied to identify people in small areas, for example home or small office with less individuals.

Acknowledgements

I would like to thank my supervisor Dr. Meysar Zeinali, School of Engineering, Laurentian University, for his clear guidance and encouragement. I would also like to thank him for his original ideas on Human Head detection. Further, I would like to thank him for his patience and for numerous revisions of my drafts.

I would like to acknowledge my friends Jianjun He, Jasmin Lemieux, Jianing Yang, Ahlam Maremi, Louis-Francis Tremblay, Mouna Haouas, and Navid Seyed Hosseini for allowing me to use their picture. I would also like to thank them for being part of my research and their valuable time. Further, I would like to acknowledge Greg Milks for making some improvements in MATLAB Program for Human detection.

Lastly, I would like to thank my parents and my family for constant support and encouragement.

Table of Contents

List of Figures

List of Tables

Nomenclature

AAM	Active Appearance Model
ACM	Active Contour Model
ASM	Active Shape Model
CNN	Convolutional Neural Network
EBGM	Elastic Bunch Graph Matching
GWN	Gabor Wavelet Network
HRI	Human-Robot Interaction
ICA	Independent Component Analysis
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LPT	Log Polar Transform
OSH	Optimum Separating Hyperplane
PCA	Principal Component Analysis
SVM	Support Vector Machine

Chapter 1

INTRODUCTION

In many developed and developing countries, the use of robot is increasing steadily in all kinds of industries and in households [1], [2]. Robots can work independently or in co-ordination with humans. Robots are used along with human so that the best capability of robots and humans can be used to perform a task effectively and efficiently. While robots can outperform humans in repetitive task, humans perform better than robots when it comes to logical thinking, collaborating in a group and responding to a sudden unexpected behavior. Thus, by utilizing the best qualities of humans and robots, a task can be performed effectively and efficiently while using as fewer resources as possible[3].

As humans work along with robots in Collaborative/Hybrid Manufacturing, for efficient execution of a task, a reliable interaction between humans and robots is of utmost importance [3]. Robots can work effectively and efficiently in coordination with other robots or humans if it has knowledge of the work floor and knowledge regarding presence of humans or other robots in its work floor. Additionally, information about location of humans helps to maintain safety in the workplace whereas the information regarding interaction with humans is very important in case of service robot.

A robot capable of identifying humans can provide better service to the end users. For service robots, more interaction is needed between humans and robots as service robots must provide specific service to specific persons. On the other hand, for industrial robots, identification of humans, may help in ensuring security of the person in need of help or service from robot.

While detection and identification of person is an easy task for human, it can be challenging for robots as many factors such as indoor lighting conditions, use of hats, long beards etc. can reduce human detection rate by computer. In this thesis, a human detection method is first proposed based on shape of head and color information is used as secondary criteria to localize head. Whereas, in Face Recognition process points around the corner of the eyes, nostrils and mouth are taken into consideration and response of Gabor filter of different wavelength and orientation on those points is used to train a classifier. When a camera detects a new person, it classifies the person into one of the person stored in the database.

1.1 Objectives

The objective of this thesis is to develop an indoor human detection and face recognition method using single camera. The goal of this thesis is to:

- Improve human-robot interaction

To improve interaction between humans and robots, an effective interaction is necessary. Computer vision system helps a robot to perceive its surrounding. By improving detection of human, a safe collaborative environment can be maintained where robots and humans can work together.

- Use curvature criteria to detect circular object such as human head

As human head has certain circular shape, curvature criteria can be checked to identify head curves. The curvature criteria can also help to identify objects with certain curvature. In this thesis, application of curvature criteria to detect circular objects and human head is investigated.

- Detect and track human presence in the surrounding

In collaborative manufacturing, human and robot work together to perform task efficiently and accurately. Both, human and robot have their strong points and weaknesses. While human can perform superiorly in terms of immediate response to external changes, robot performs better in performing repeated task with precision. As, robot is performing its task it should also be able to acknowledge human presence in its surrounding. Thus, the main aim of this thesis is to first identify human presence using computer vision. Once human is identified, human is tracked using the detection criteria on each frame.

- Recognize the person in vicinity of a Robot

In a workplace, different functions are assigned to different person. While human can know the function of different person by identifying their face, name tag or uniform easily, it might be difficult for a robot to identify them. Thus, facial recognition method can help them to identify the person and help/collaborate with the identified person. Further, face recognition method can also be crucial for security purpose.

1.2 Thesis contribution:

The contribution of this thesis is to develop an algorithm for human recognition in indoor environment and to identify them.

- Develop an algorithm using shape and color information to detect human in indoor environment.
- Identify the persons on field of view of camera once the camera detects them.

1.3 Thesis outline:

The thesis is organized as follows:

Chapter 1 provides the introduction of thesis topic, the objective of thesis and contribution of this thesis. It further provides an outline of the thesis.

Chapter 2 provides overview on the advancement made in human detection and face recognition methods. In this Chapter, human detection method using single and multiple sensors is presented. Further, different face recognition method in still image and video sequence is also presented in this chapter.

Chapter 3 presents a brief background on topics covering digital image processing, human detection, human head detection and face recognition. This chapter is intended to provide the reader with sufficient background on human detection and face recognition algorithms.

Chapter 4 presents the proposed human detection method. In this chapter, the shape information of human head is obtained by calculating the curvature of human head. After finding, possible location of human head, face position is approximated by skin color detection. By combining shape and color information, final possible location of human head can be determined.

Chapter 5 presents a method to identify different person once they are detected by the method presented in Chapter 4. In this chapter, it is investigated if only Gabor filter response from nine face keypoints would be sufficient to identify persons on a small database.

Chapter 6 concludes the thesis with contribution of this thesis and recommendation for future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides a brief overview on challenges and progresses made in human detection and face recognition methods. The aim of this chapter is to provide readers information regarding basic concepts and advances made in the field of human detection in indoor environment and face recognition process. First, literature review on human detection method is presented and then literature review of face recognition is presented.

2.2 Human-Robot Interaction

The use of robot is extending from the traditional industrial work floor to collaborative manufacturing environments (i.e., hybrid automated manufacturing) [3], living and working environments such as use of service robots in health care sector [4], e.g. Hybrid assistive limb also called robot suit [5], or as a more personalized appliance such as robot vacuum [6]. The use of autonomous professional service robots, working side by side with human and collaborating with human is expected to grow in the coming years [7] and it requires human-robot interaction (HRI). HRI is also critically important for advanced robots, particularly those with potential applications in hybrid automated manufacturing systems to achieve the desired efficiency, productivity and safety [8]. HRI as a field is advancing very fast and thus it continuously demand new and effective method of assessment and evaluation of interaction between humans and robots [9]. Interaction, which involves two or more parties working together to complete a task, depends upon how a team is formed, the level of desired autonomy and the method of

information exchange [10]. For instance, humanoid robot Asimo can identify different person, accept human voice as a command, and perform tasks like serving drinks to the people while avoiding collision with the help of vision technology.

For HRI, designing of technologies and evaluation of the capabilities of robots on using these new technology is essential [10]. For effective HRI, robots should be able to communicate and their communication capability depends both on their appearance and attributions [11]. In robots such as ECCEROBOT, biological inspiration can be seen not only in appearance but also in mechanism and behavior of the entire robot [12]. ECCEROBOT is equipped with biomimetic technology with help of exteroceptive sensors (visual and haptic) which greatly helps the robot to interact with humans [13]. Further to facilitate HRI, it is suggested in [14], that sensors which imitates humans sense organs such as video cameras representing eyes and microphone representing ears could be used. The vision system linked to a robotic system helps to detect a person or an obstacle and thus is important for robot's perception of its surrounding. In this thesis, a human detection method is proposed using single camera to facilitate HRI.

2.3 Human Detection

For both service robot and industrial robot, its workplace is shared with other robots or humans thus to avoid collision and to improve collaboration it needs to interact with humans or robots [15]. For interaction, detection of other robots or humans is the first step. Vision and acoustic information are vital for human detection [16]. Human detection by computer vision system linked to a robotic system however still remains a challenge due to wide variability in the appearance due to factors such as clothing, make-up, lighting condition and so on [17]. The human detection problem is briefly explained next.

2.3.1 Human Detection problems

The objective of human detection is to find all the people in an image plane [18]. Color and shape of the human face and head can be regarded as distinct feature for detection of human. However, there might also be other objects having color or shape similar to human head and face along with humans in indoor environment. Thus, detecting a human and tracking under complex conditions such as dynamic and cluttered environment can be a challenging task and has not been well-developed area, therefore needs more research. The existing human detection method is presented next.

2.3.2 Human detection Classification

Investigation of relevant literature shows that [19], human detection method is broadly classified into:

> ***i***) Background subtraction, where the foreground object is found and labelled into different groups such as human, objects, vehicle etc. based on their color, shape, motion or other features.

> ***ii***) Direct techniques based on features extracted from image or video patches; the features include shape (in the form of contours or other descriptors), color (skin color detection), motion or combination of two.

Some human detection method based on these two types is presented next. Background subtraction and k-means clustering has been used for detecting human in [20] with 60-70%

success rate in detecting human in indoor environment. In reference [20], before applying k-means algorithm, the input image is converted to LAB color space (where L stands for Lightness and a and b are color opponent dimension), as LAB color space can segment skin colored object better than Red-Green- Blue (RGB) color space. It was found that three clusters give the best result and the accuracy, and the processing time is increases with the increase in number of clusters. Though this concept can differentiate skin from other colored object, it might fail to detect skin when skin colored object is present. If error rate is to be reduced, the number of clusters should be increased which increases the computational time.

Another human detection algorithm using background subtraction consists of foreground segmentation, human recognition, human tracking and false object detection [21]. The mixture of Gaussian, Nonparametric kernel and codebook used in reference [21] can have better performance in human detection but it needs extra computation. For human detection, codebook is used to differentiate human from other objects. First the size of the object is normalized to 20 by 40, and then the shape of the object is extracted as feature. On second step, feature vector is matched with code vectors of codebook which has minimum distortion. If the minimum distortion is less than threshold, the object is classified human. Although, background subtraction works reasonably well when the camera is stationary and the change in ambient light is gradual, it is sensitive to changes of dynamic scene due to lightening and extraneous events.

An alternative method for human detection is the method using direct technique based on features (shape or color) extracted from the input image. In [22], it is claimed that skin detection and segmentation is popular and useful technique for detecting human and human body parts. However, it is also mentioned that changing lighting conditions and complex backgrounds are major problem of this system limiting their use in practical real-world application. Their fuzzy

skin detection approach has two parts: ***i***) Pixel wise classification, ***ii***) Controlled diffusion. According to reference [22], in fuzzy pixel wise classification a value of 1 is assigned to the center of the cluster and it decreases to 0.5 on the border of the cluster. The final decision about assigning the class of the pixel is made with the context information. Whereas in controlled diffusion, a given pixel is a member of skin class, if its Euclidean distance with one of its direct diffusion neighbor, already labelled as skin class, is smaller than a certain threshold. Although this method is a good approach for detecting whether a cluster has skin color or not, it might not differentiate human skin color from a furniture having similar color.

Reference [23] combines the output of two different modules; one that matches the intensity gradient along the object's boundary and one that matches the color histogram of the object's interior. In the paper, human head is modelled as a vertical ellipse with a fixed aspect ratio of 1.2. Like other skin detection method, the method may fail to detect skin when lightening condition changes or when camera gain is adjusted automatically by the camera. While color detection can be a very good method to detect presence of human, people have different face color which makes it challenging to develop an algorithm that works for wide range of skin colors. Local binary pattern operator and moments is used for Human detection in [24]. Local binary pattern operator for different number of pixels and different radius in calculated. It is reported that moments can be used for binary or gray level description. Further it is reported that moments provide an interesting and useful way of representing shapes and can be used detection of human in image. Support Vector Machine was used as classifier in the study to differentiate between Human and Non-Human objects in real scene. The drawback of this method is that this method is slower compared to some of the existing methods.

In [25], it was found that the appearance of skin color changes as the person moves toward or away from the camera or when light changes. It is concluded that the ability of handling light changes is the key to success for a skin-color based human detection. Thus, it is shown that intensity is the main difference and not the color itself. However, differentiating between pixel representing human skin and pixel of other object having skin color is still a challenge. It is thus seen that though the accuracy in detection of human is increasing but it is still not accurate enough. From the preceding paragraphs, it can be concluded that combination of two attributes i.e., color and shape information can be more successful in human detection.

In [26], the performance of human detection by indoor robot was investigated using different algorithms; object segmentation from depth images, convolutional neural networks and fusing the results. The performance was evaluated using three different sensors; structured light sensor, stereo camera, and Time-of-flight camera. It was shown in this study that combination of two different algorithms outperformed algorithm using only single method. Considering complexity of human detection task, many current approaches for human detection uses single or multiple cameras along with other sensing device as thermal camera, depth camera, LASER, Infrared sensor to detect human [27]–[29]. Using multiple sensors makes those methods financially as well as computationally expensive. Complementary advantage of using visual and thermal information is used in [29]. In the paper, two main processing steps are used for human skin detection; pixel-wise classification and spatial diffusion. Pixel-wise classification uses a non-parametric skin model implemented using histograms and spatial diffusion takes neighborhood information into account when classifying a pixel. Human blob detection uses thermal information for detection of human body parts. A probability ratio is used for detection of thermal human skin. The threshold however is adapted to account for changes in temperature of

the surrounding and changes in response to camera. These methods require high cost thermal sensor which are not cost effective. Additionally, the response is dependent on how long the camera is on because image becomes saturated after some time and thus new threshold needs to be adjusted [29]. Some researchers have combined Laser technology and computer vision in their research [28]. In the paper, laser range finder is used to detect and track human and then vision information is used to identify segmented person. This method uses a collection of self-organizing maps rather than a single neural network. However, any moving object could be considered as human in this method.

One of the human detection method [27] consists of multi sensor data fusion technique to integrate the following two source of information; the first one is leg detection based on the laser scans of SICK Laser range finder and the other is face detection which uses a monocular camera. Human detection by leg detection however might not work for detection of human sitting on wheel chair. In [30], the 3D position of human's head and face is estimated by using depth from focus method. For a person who is 90 cm to 340 cm from the camera, the system can measure the head location with less than 10 cm error. This method detects human head and face by using motion detection, Hough transform and statistical color model. By changing pan, tilt and zoom, the human face is placed in the center of the camera. To detect motion, temporal derivative of an image is calculated and thus image can be differentiated into stationary and moving part. It is stated that head can be supposed to be a nearly rigid body and its shape in an image can be approximated by a rigid two-dimensional model. To find head in the image, the pixel location from the topmost part of the image is analyzed first. Skin color detection is only used on the position of human head. The limitation of this approach is that the person needs to be almost static which may not always be the case.

While most of the human detection method has focused on detection of human as a whole, some method has also focused on detection of human by detecting head. As human head has fixed shape, detection of head to detect human can be a good approach. The existing methods of head detection can be divided into three main kinds, i.e. methods based on color model, methods based on template matching and methods based on contour detection [31]. Method based on color mode is simple and have low computational cost. However, when the color of hair or skin changes significantly they often fail to detect head. Method based on template matching detects regions belonging to head by measuring the similarity between a head template and search area. The head templates according to them is defined in terms of geometric primitives such as lines, polygons, circles, and arcs, and fitness criteria is employed to degree of matching. Method based on contour detection skillfully uses some geometric curves to describe head and shoulder information. The fitting process can be very time-consuming in this process.

An improved moving object detection algorithm has been used in one of the paper to detect head and start tracking progress [31]. The head detection and tracking algorithm is based on skin color and head shape model. Omega shape based method is generally used for ellipse detection. The foreground is first extracted using an improved Gaussian Mixture Model approach. Then a new Gaussian energy-based seed-growing algorithm is used to locate rough position of head and shoulder. After finding rough position, the exact position of the head is obtained by fitting ellipse on the head contour. A major drawback of this method is that it is susceptible to the influence of outliers, noise and extraneous pixels.

Head detection is more difficult than face detection as head may be under any arbitrary pose, head detection should work for a wider range in comparison to face detection and head detection should work on real time which increases the computational complexity [32]. In the paper [32], a

real time elliptical head contour detection method based on quadrant arcs is proposed, which was found to be efficient for arbitrary head pose and wide distance range. The connected edges are detected, discriminated, and combined to fit out the elliptical head contour in the program. The process is claimed to be robust to detect person with long or short hair, and person with or without hat. The drawback of this system is extra calculation needed to split an image into different arc quadrant, combine them and finally use of ellipse fitting method for head detection.

Some application may also demand recognition process after detection of human. This might be useful in industrial application where certain person in authority should be given access to enter a workplace and stop other from entering there. Thus, face recognition combined with human detection method will be of even more value to the end-users.

2.4 Face Recognition

Once, it is known that there is possible human in the surrounding of the robot, the next step is to recognize them. Recognizing a person helps a robot to provide person specific service or grant access to restricted areas in industry for authorized person only. Face recognition result can't be altered, is non-intrusive and can be performed with less expensive camera and has thus emerged as a reliable biometric-based technique for person identification [33].

Face recognition is basically used for two purpose; ***i***) Verification (one to one matching), where an input image is checked to see if it matches the claimed identity. ***ii***) Identification (one to many matching) where an unknown input image is identified by comparing with the known images in face database. Early face recognition system considered face recognition system as one of the object recognition problem. However, as compared to rigid object recognition, there may be many factors such as light, facial expression, facial make-ups etc. which makes the face

recognition problem difficult. A block diagram of a face recognition system is as shown in **Error! Reference source not found.**.

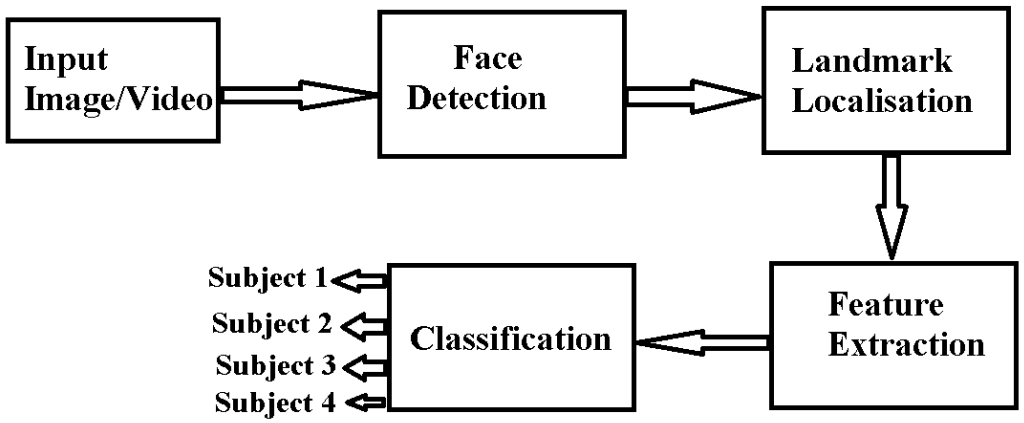


Figure 2.1: Block diagram of a Face Recognition System.

2.4.1 Face Recognition Problem

The general statement of problem for face recognition can be formulated as follows: from the input image or video sequence, identify or verify the person with reference to stored database of images [34]. The same conditions affecting human detection process i.e. lighting, make-ups and so on also affects face recognition process. The challenges in automatic face recognition system [33] are as follows:

a. Ambient lighting condition may be different and may affect the recognition rate.

b. The person may face camera from different angles or may not face the camera at all.

c. Face may appear different because of glasses, beard and hair style.

d. The appearance of face also varies with age and may change with expression.

Thus, in many Face recognition method various assumption are made for the recognition task. While notable success has been achieved in face recognition from still images, face recognition from real time video is still an ongoing research.

2.4.2 Classification of Face Recognition method

While most of the earlier face recognition was performed in still images, currently most of the systems perform face recognition from live videos in real time. Further, as Webcam is used in this thesis, literature on 3D face recognition is not presented in this thesis. Face recognition method in general can be broadly classified into [34]:

a. Face recognition from still images

b. Face recognition from videos

First some studies done on face recognition from still images will be presented and then face recognition from videos will be presented.

a. Face recognition system from still images

Figure 2.2: One of the Subjects from AT&T Database [35].

The study of face recognition started with face recognition from still images. In those images, the image would be normally pre-processed so that there won't be much variation in the lighting condition and the facial expression was controlled. **Error! Reference source not found.** shows one of the subjects from AT&T [35] database which was used to train and test the accuracy of face recognition in still images. The background is uniform and there isn't much variation in the lighting condition or facial expression in these images. Face recognition system from still images can be further classified into two types:

1. Feature based System

2. Holistic System

First some of feature based methods will be explained and then some holistic method for face recognition from still images will be explained.

1. Feature based System.

In Feature based system, first step is identification of different facial parts such as eyes, mouth and nose. Second step consists of constructing a feature vector which consists of direct measurement of these parts such as eyes, nose, mouth etc., or ratio between different parts or other relationship between them [34]. After constructing the feature vector, the recognition task consists of comparing one feature vector of an input or test image with other feature vectors stored in the database.

One of the earliest face recognition method [36] has used 16 geometrical features such as breadth of mouth, breadth of eyes and so on to recognize a person. This method had an accuracy of 70% in a database of 20 persons. In this simple two step method, first eyes, nose and mouth are

localized manually and next the corner points in these organs are detected. During that time, research was also focused on automatic systems capable of segmenting objects [37] or findings contours [38] which could be preliminary step in automatic key-points localization. Before Deformable Templates [39] were used to fit in a model of eye and mouth in the image, Active Contour Model or 'snakes' [37] had been used to find contour of different objects. This concept has been extensively used and other features such as shape [38] and shape along with texture [40] has been used which has greatly facilitated automatic localization of different facial parts.

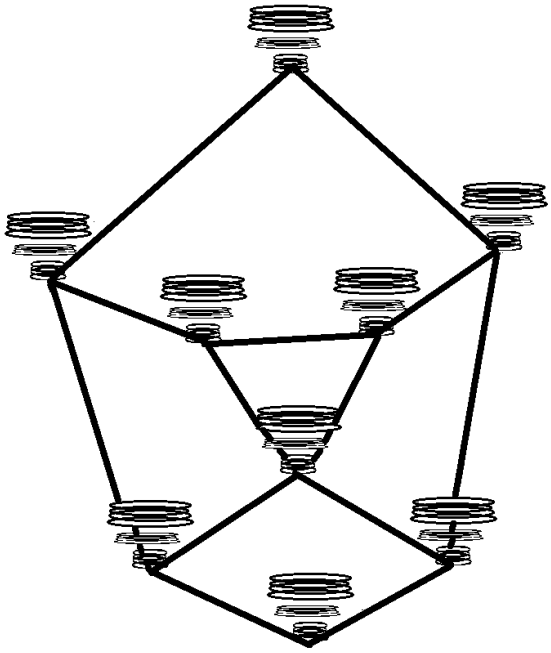


Figure 2.3: Image Graph (reproduced from [41]).

In another most cited literature [41], Gabor filter response on different fiducial point is used to form an image graph as shown in **Error! Reference source not found.**. The graph of a new image is compared with face graph of all the images in the database for recognition process. Recognition rate up to 98% was reported with this elastic bunch graph matching (EBGM) method.

It is seen from the above studies that for feature based face recognition system, the localization of the key points is the preliminary step. Following concept of [40] and [41], many studies have been performed to locate facial key points in previously unseen, so called 'images in the wild', in [42]. Additionally, the method presented in [42] could also estimate the pose of different subjects after detecting face. Automatic landmark localization has been on-going research [43] and algorithm which can accurately localize key points will greatly improve the accuracy of feature based face recognition system.

2. Holistic system

In holistic approach, the recognition process is based taking into consideration the whole image of the face. If two images have the same size, a simple method to compare them could be to compare all the gray level intensity present in those images. Although, PCA was used first for data compression [44], this concept was used for face recognition in [45] for the first time. This Eigenface based system greatly reduced the dimensionality and the recognition could be done in a space having much lesser dimension than the original one. If there are 20 persons on a database, then for an image of 128X128 pixel, the calculation can be reduced from a 16,384 dimension to 20 dimension, which will be quite manageable in terms of processing. This method [45] is considered as the first automatic facial recognition method and has been widely studied and many variations [46]–[49] has been proposed. It was however shown later in [50] that Fisherface based approach using Linear Discriminant Analysis (LDA) had lower error rate than Eigenface based approach. The Fisherface based approach is reported to be insensitive to both lightening direction and facial expression. It is reported in [50] that although PCA based approach can be a good option for image reconstruction using low dimensions, it is not ideal from discrimination point of view. A good comparison between these PCA and LDA based

method on different conditions can be found in [50]. In the following years, different modification of eigenface and fisherface [51]–[55] has been proposed for face recognition. Other subspace based recognition method such as Independent Component Analysis (ICA) [56] and face recognition using Laplacian face [57] has also been proposed. It was shown in [56] that ICA outperformed PCA.

Later, the holistic approaches have been used along with Neural Network as in [58]. Neural network can perform complex functions such as pattern recognition, identification, classification, speech, vision and control system [58]. Neural network develops compact internal representation of faces which is similar to feature extraction process. Most of the information regarding the input is stored in hidden neurons, whose number is much less than the input or output [59]. Neural network has been widely used in pattern recognition and a full review of application of artificial neural network (ANN) can be found in [60]. Deep learning [61] and in particular convolutional neural network (CNN) [62] has achieved promising result in face recognition recently. It achieved a recognition rate of 88.7% in LFW (Labelled Face in Wild) database [61]. One of the recent face recognition method using CNN developed by Google Incorporation was trained using 200 million images and eight million unique identities [63]. Recent face recognition systems [63]–[66] have used many variations of CNN for face architecture. A method of face recognition was developed using convolution network in [67]. The network contained ten layer and last layer was used as input for training. The similarity between two image was measured using L2 norm. A recognition rate of 95% was reported using the method on Labelled Face in the wild database. Further, the result was investigated on real world application and it was concluded that the method didn't meet the requirement of real world application.

When different new approaches were taken for face recognition from still images, the accuracy went on increasing [68]; even for images in the wild (i.e. images could have wide variation in ambient lighting condition, facial expression and so on.). At this point, researchers were interested in face recognition from video sequence as it could be beneficial for security purpose. Face recognition from video sequences is explained next.

b. Face Recognition from video sequence

Face recognition from video provides both challenges and opportunities. The challenge is that real time implementation could be difficult to achieve considering huge amount of data to be processed in real time whereas the opportunity is availability of abundance of information regarding face. In video-based face recognition, the gallery set (database) and the test set is both a collection of sequential images rather than a single image [69]. Methods based on Image sets [70], [71] has higher recognition rate than the one based on single image because they contain information about test image with different pose and under different lightening conditions [72]. Video based face recognition can be generalized to image set based classification [69].

In [73], Image sets are modelled as Gaussian Mixture Model. Although it is mentioned that the dissimilarity between two image sets can be calculated as distribution divergence in their Gaussian Mixture Model, it is also mentioned that distribution divergence may not be discriminative enough, considering wide variation within single image set. Thus, an efficient method of representing Gaussian distribution had to be developed. A discriminative analysis method on Riemannian manifold was developed and the performance was checked for both verification and identification task. The method was reported to have superior performance compared to other state-of-the-art methods.

The idea of face recognition from still images has been extended to face recognition in videos in [74]. At the training stage, the faces from same person are grouped into clusters with K-means algorithm, and the cluttered faces from different persons are further grouped according to their semantics. Then, the features are further divided into '*n*' slices and PCA and LDA is applied on those individual slices. In the assigning stage, first the matching score of the input image with the database frame is obtained and finally the best fit is found in the corresponding subspace. In the method, the persons in video are asked to read number from zero to nine. From each video sequence, 21 frames were selected where each frame corresponded to the peak of sound wave while reading those numbers. They have shown that superior performance in face recognition could be achieved by using some key frames instead of treating all the frames in video equivalently. As it was seen that multiple sensors could be used to detect human, similarly information from various sources can be combined to develop a face recognition method. On the other hand, in [75], information from the text (scripts) and subtitles are used to automatically assign a name to a character appearing in the video. By doing so enough information about the speaker, text and the time frame is gathered. The identity suggested by the text and subtitle won't be sufficient to prove that the person suggested is the one appearing in the video. To identify the possible person, person whose lip's motion is constantly changing is identified as speaker. In their, matched background similarity model for recognizing face, Local binary pattern (LBP) was used for feature extraction while Support Vector Machine (SVM) was used as classifier.

A video based face recognition system is presented in [76] which uses skin color model to detect face and Gabor wavelet network (GWN) to detect facial landmarks. Tracking was performed

using GWN and recognition using eigenfeatures selection. The main idea of this method is that if the appearance manifold 'M_k' for each individual 'k' is known, then the identity 'k*' can be determined by finding the manifold 'M_k' which has minimum distance to input image 'I', i.e.[76],

$$k^* = \arg\min d_H(I, M_k) \quad (2.1)$$

Here, 'd_H' denotes the L2-Hausdorff distance between input image 'I' and manifold 'M_k'. As discussed in feature based face recognition from still image (i.e., Section 2.3.1.a), automatic localization of landmark points is the key factor for face recognition. Color information only can also be considered as the first step for face recognition. In [77], a skin color model is first used to detect face. Then, the pupils are detected while searching for two large blobs then corners of the lips and nostrils are found using the approximate position of the pupil. After detection of the key points, the face region is cropped, resized and registered. After image registration, a two-step process is applied for illumination normalization. The face recognition process consists of calculating inter manifold distance between test set and database set. Although, a good recognition rate of 95% is reported in this method, it is also mentioned that the training and test set consists of the same sequence.

2.5 Summary

In this chapter, a detailed review of recent studies related to this thesis was provided. Different method of human detection and face recognition was presented. Further, the challenges that remain in human detection and face recognition was briefly presented. It was seen that concept used in recognition of person in still images has been used in recognition of person from real

time videos. Next, chapter three describes relevant theory and background related to this research.

Chapter 3

THEORY AND BACKGROUND

3.1 Introduction

In this chapter, relevant theory and background related to human detection and face recognition method used in this thesis will be discussed. As, the human detection method is based on curvature calculation, first in section 3.2, theory on curvature and curvature calculation is presented. Then, section 3.3 presents theory on Automatic landmark localization which is the most important step for feature based face recognition. Further, section 3.4 presents theory on Gabor filter, section 3.5 presents theory on feature selection and section 3.6 presents theory on Support Vector Machine. Finally, section 3.7 and 3.8 presents some human detection and face recognition method respectively and section 3.9 concludes the chapter.

3.2 Curvature

Curvature feature hasn't gained enough attention in comparison to other features; such as color, orientation, motion, size, depth, Vernier offset, and gloss in visual search [78].

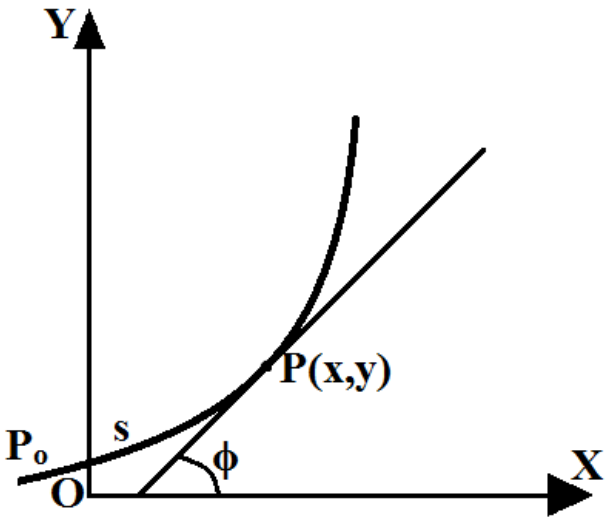


Figure 3.1: Curvature of a curve (reproduced from [79]).

Thus, the use of curvature to segment out curves having certain property is investigated in this thesis. The curvature (κ) of a circle is simply the reciprocal of its radius '*R*' i.e.,

$$\kappa = \frac{1}{R} \ . \tag{3.1}$$

On a given curve as shown in Figure 3.1 the angle 'ϕ' from the x-axis to the tangent at the variable point *P*(*x*,*y*) can be considered as a function of the arc *s* from a fixed point *Po* of the curve to *P*. Moving along the curve, 'ϕ' measures the change in the direction of the tangent, and the derivative of 'ϕ' with respect to *s* is change of the direction per unit distance moved. This derivative is the curvature at *P* [79].

Mathematically,

$$\kappa = \frac{d\phi}{ds} \ . \tag{3.2}$$

where, ϕ is the tangential angle and *s* is the arc length.

In a circle with radius *R* as shown in Figure 3.2 [79],

$$\phi = \theta + \frac{\pi}{2} \quad . \tag{3.3}$$

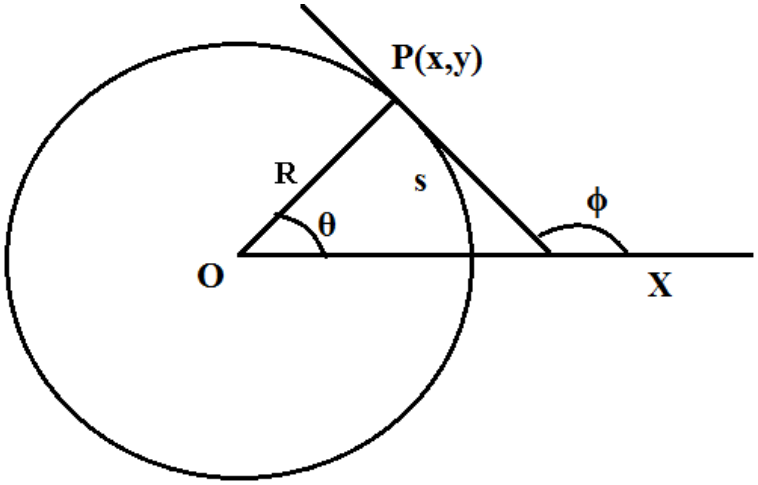


Figure 3.2: Curvature in a circle (reproduced from [79]).

Hence, the curvature is,

$$\frac{d\phi}{ds} = \frac{d\theta}{ds} = \frac{1}{R} \quad . \tag{3.4}$$

For other curves, the curvature is approximated by assuming that there is a circle which overlaps a certain section of the curve. In Figure 3.3, for a curve *C* at point *P*, there is a circle which touches certain portion of it. The curvature of that portion of the arc is found by using the radius of the circle *R* [80].

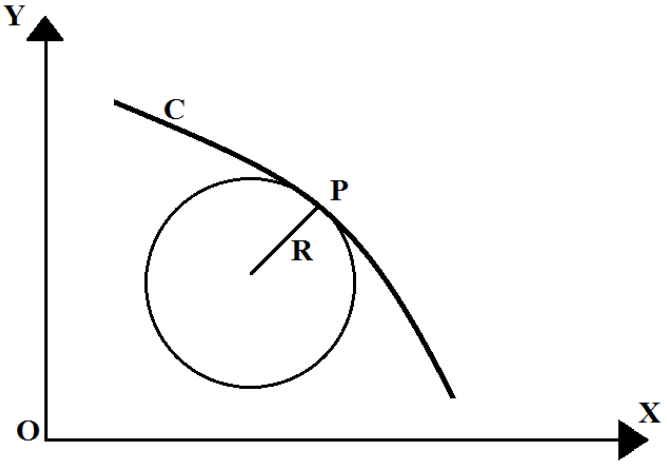


Figure 3.3: Curvature of an Arc

The approximation of curvature in a curve can be done by taking three points supposed to form a triangle on the circumference of the circle as shown in Figure 3.4 [81]. From the corollary of classical theorem of elementary geometry, the circumscribed radius is found by dividing the product of sides *a*, *b*, and *c* of a triangle, by four times the area (Δ) of the triangle. Mathematically,

$$R = \frac{abc}{4\Delta} \quad . \tag{3.5}$$

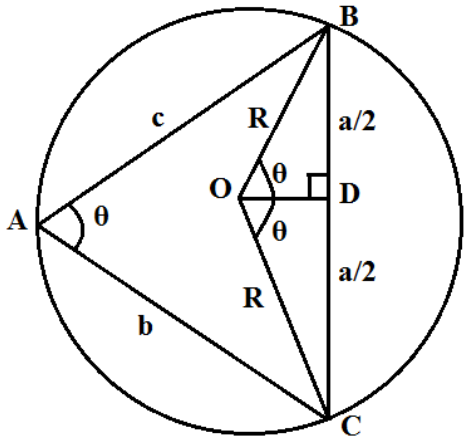


Figure 3.4: Curvature of three points on the circumference of a circle.

For its proof, a triangle ABC as shown in Figure 3.4 is considered to be circumscribed in a circle having radius *R*. As the central angle is twice the inscribed angle, ∠BOD = θ.

In triangle BDO,

$$\sin\theta = \frac{\frac{a}{2}}{R} = \frac{a}{2R} \quad . \tag{3.6}$$

Now, the area of triangle ABC ' Δ ' is given by,

$$\Delta = \frac{1}{2} bc \sin\theta \cdot \tag{3.7}$$

Substituting value of $sin\theta$ from Equation 3.6,

$$\Delta = \frac{abc}{4R} \ . \tag{3.8}$$

Hence, $R = \frac{abc}{4\Delta}$.

Then, from Equation 3.1, $\kappa = \frac{4\Delta}{abc}$. (3.9)

Let A(x_1,y_1), B(x_2,y_2), and C(x_3,y_3) be the three points on a triangle ABC. The area of a triangle with three points can be calculated as follows,

$$\Delta = \frac{1}{2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{3.10}$$

or, $$\Delta = \frac{1}{2}\left|x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)\right| \tag{3.11}$$

The above equation can be further simplified to numerator of Equation 3.12. The curvature in 2D image drawn through three points can be approximated by substituting value of Δ from Equation 3.11 in Equation 3.9 i.e, $\kappa = 4\Delta/abc$ as:

$$k = \frac{2\left|(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)\right|}{\sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)((x_3 - x_1)^2 + (y_3 - y_1)^2))((x_3 - x_2)^2 + (y_3 - y_2)^2)}} \tag{3.12}$$

By using curvature criteria, only relevant curves can be retained for further processing. On those curves which are retained, circle fitting can be done and skin color can be checked as secondary criteria for detecting human head.

3.3 Automatic Landmark Localization

From Chapter 2, it is seen that there is holistic method of face recognition and feature based method of face recognition. Feature based recognition method are getting more popular because of its capability to represent the information of the face in a more compact way.

The concept of using particular discriminative points, referred to as fiducial points, for face recognition was used in [41] as preliminary step for face recognition. The landmark localization process in that paper was semi-automatic. Gray scale information was used in many of those researches then [82]–[84] to localize facial key points. Semantic knowledge of face is used in [83] to localize 18 facial feature points with a good average accuracy of 90% on different database. Although, use of gray level information only gave some promising result it was obviously difficult to get constant reliable detection and localization result under varying lighting and pose changes. Also, while detection of some of the points such as iris can be relatively easy, finding corner of mouth can be difficult. Additionally, the result can be also affected to some extent by length of hair, use of scarf, glasses and so on.

Before different key points could be localized with accuracy, it is necessary to find the boundary of face as preliminary step. Application of 'Snakes' [37] has been used in [85] to find the face contour [86]. The idea of 'Active Contour Model' or 'Snakes' has been studied and updated with other features which resulted in 'Active Shape Model' [38] and 'Active Appearance Model' [40] which serves as starting point of many automatic facial key points localization methods available today. The main idea of 'Active Shape Model' was that a model should be allowed to deform only in a way that is learned from its training set. Thus, every landmark which represented the shape of the body could move only to the extent that is learned from the training set. Thus, by

applying Principal component analysis (PCA) to the data set, the variation of shape of an image could be learned. A brief overview on application of PCA is presented next. PCA starts with the calculation of the mean shape (μ_{shape}) [38]. If $x_i = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ represents coordinates of n points which represents the shape of an image then for No number of aligned shape, the mean image (μ_{shape}) is calculated as :

$$\mu_{shape} = \frac{1}{N_o} \sum_{i=1}^{N_o} x_i \quad . \tag{3.13}$$

For each shape 'x_i', its difference from the mean shape '$diff_{xi}$' can be calculated as:

$$diff_{x_i} = x_i - \mu_{shape} \quad . \tag{3.14}$$

The collection of difference shape '$diff_{xi}$' can be used to generate a covariance matrix 'S_x'.

$$S_x = \frac{1}{N} \sum_{i=1}^{N} diff_{x_i} diff_{x_i}^{\ T} \tag{3.15}$$

From the covariance matrix, eigenvalues, and eigenvectors (λ_i ,P_i) are calculated. The eigenvectors P_i gives the most significant mode/direction of variation of the data whereas the eigenvalues gives the weight of such variation. Most of the variations in the data can be explained by small number of modes t [38]. Thus, any shape in the training set can be calculated using the mean shape and the eigenvectors.

$$x = \mu_{shape} + Pb \tag{3.16}$$

Where, $P = (p_1, p_2, \ldots, p_n)$ is the first 't' eigenvectors and $b = (b_1, b_2, \ldots, b_t)^T$ is a vector of weights. The value of k_{th} vector weight b_k is suggested as follows:

$$-3\sqrt{\lambda_k} \le b_k \le 3\sqrt{\lambda_k} \tag{3.17}$$

Where, λ_k is the k^{th} eigenvalue.

By changing the values of b 'shape parameters', a shape X can be deformed to fit into target M_Y [87] which is the main idea of Active Shape Model. The algorithm to fit a model to a target can be summarized as [87]:

1.Intialize $b = 0$.

2. Generate model points

$$X = \mu_{shape} + Pb.$$

3.Find scaling factor 's', rotation factor 'θ' and translation 't' which best fit M_Y to X.

4. Project M_Y into X-space to get 'm_y'.

5. Update model parameter.

$$b = P^T(m_y - \mu_{shape})$$

6. Go to step 2, iterate until convergence.

The coordinates of M_Y is determined by finding the point having strongest gradient magnitude in the normal to the position of mean shape coordinates. Thus, the algorithm of Active Shape Model can be summarized [87] as below:

1.Intialize $b = 0$.

2. Find the coordinates M_{Yi} from the image searching for strong edge in normal direction of X_i.

3.Find scaling factor 's', rotation factor 'θ' and translation 't' which best fit M_Y to X.

4. Fit the new parameter to M_{Yi} to get update model parameter b. Enforce constraint

$$|b_i| < 3\lambda_i$$

5. Iterate

While in the above algorithm, only shape information is taken into account, the Active Appearance Model (AAM) takes also the gray level information into account. It was seen from Equation 3.16 that any shape *x* could be modelled from mean shape ' μ_{shape} ' and eigenvectors P_s and vector weights b_s:

$$x = \mu_{shape} + P_s b_s \tag{3.18}$$

Similarly, gray level model 'g' of any subject can be modelled from mean gray level model ' μ_{gray} ' and eigenvectors 'P_g' and vector weights 'b_g' [40] as

$$g = \mu_{gray} + P_g b_g \tag{3.19}$$

To check the correlation between 'shape' and 'gray level', a further PCA was applied in [40]. First, a concatenated vector *b* is obtained as:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s P_s^T (x - \mu_x) \\ P_g^T (g - \mu_g) \end{pmatrix}. \tag{3.20}$$

Where, W_s is the root mean square change in *g* per unit change in vector weight of shape b_s. A new model is obtained as:

$$b = Qc. \tag{3.21}$$

Where *Q* are the eigenvectors and *c* is a vector of appearance parameters controlling both the shape and gray-level of the image. Thus, by using similar iterative scheme defined above to fit a model into a shape, a model can be fit into the input image by taking into account the mean gray level appearance and mean shape. For details, reference [40] can be referred. AAM models texture and shape information inside the objects boundary. To train AAM, images and landmark annotations of facial features is required [43]. Considerable research has been done in automatic Facial key point localization. In [42], it was shown that localization of the landmark in the wild

could be achieved in previously unseen image. A detailed study of face land marking technique can be found in [88]. In one of the recent Facial landmark detection and tracking challenge [89], it was concluded that there is significant room for improvement in the process. As, Landmark detection and tracking algorithm still can be improved, face recognition in videos can be further improved if there is improvement in detection and tracking of face landmark points. Figure 3.5 shows Landmark localization on one of the subjects of AT&T database [35].

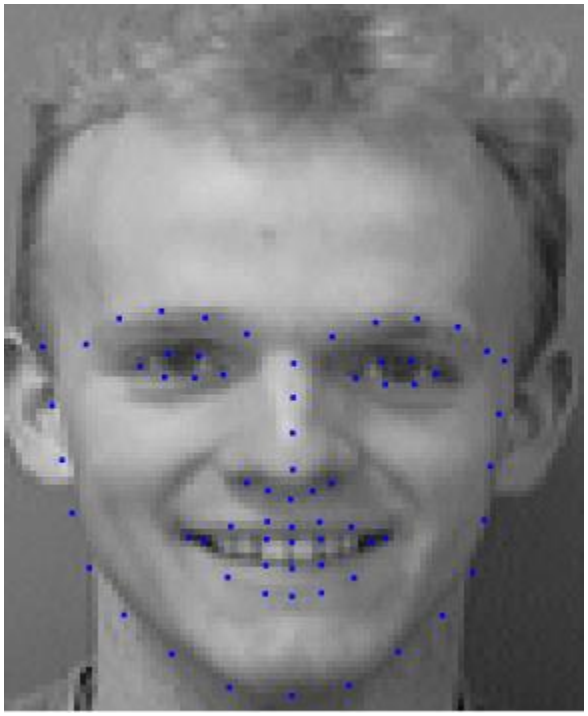


Figure 3.5: Landmark Localization [35] using "AAM optimization in the wild" [43].

3.4 Gabor Filter

Psychophysics and Physiology of human vision evidence shows that the retinal image in human eyes is decomposed into multi-frequency channels [90]. It was found that response of a class of cell in mammalian visual cortex called ‘Simple Cells’ were sensitive to frequency and orientation of visual stimuli [91]. In [92], it was shown how 2D Gabor functions could be used to

model Mammalian visual system. Gabor Filters produces multi-directional and multi-scale representation of input image which is used in object recognition task. Gabor filters have been used in object detection, face detection, face recognition as well as in facial expression recognition. Gabor function is a 2d Gaussian function multiplied by a complex sinusoid function [93] . It can be expressed mathematically as [93]:

$$g(x,y) = f(x,y)s_f(x,y) \tag{3.22}$$

Where, $f(x,y) = \frac{1}{2\Pi\sigma^2}\exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}$ is the Gaussian function and σ is the standard deviation of the Gaussian function and $s_f(x,y) = \exp\left[-j2\pi(Ux+Vy)\right]$ is the complex sinusoid function and *U* and *V* represents frequency of the complex sinusoid. The Gabor representation of an image *h*(*x*,*y*) is obtained by convolving the image *I*(*x*,*y*) with family of Gabor Kernels *g*(*x*,*y*) [94].

Mathematically,

$$h(x,y) = I(x,y) * g(x,y). \tag{3.23}$$

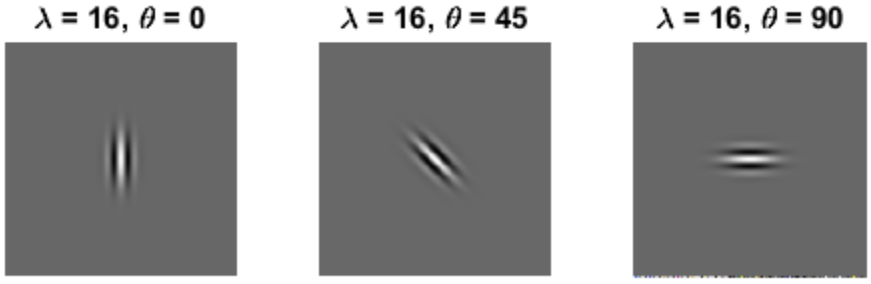


Figure 3.6: Change of real part of Gabor Kernel with Orientation.

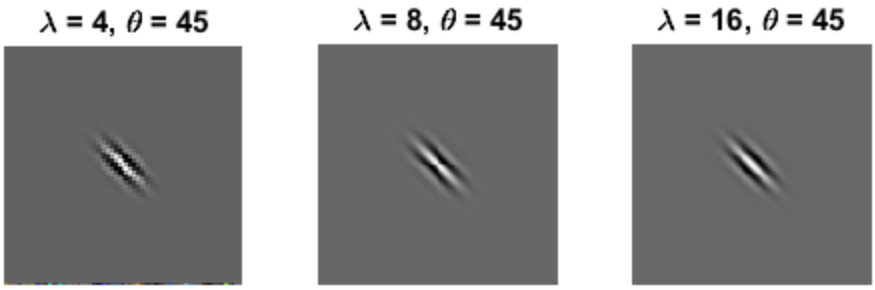


Figure 3.7: Change of real part of Gabor Kernel with Wavelength.

Gabor filter response around eye has been used in [95] to identify person with 85% recognition in AT&T database. It should be noted that images in this database has uniform background. A 25 $\times$25-pixel window was selected around the eye center found manually to form a 625-dimensional feature vector. It was found to give 85% recognition rate compared to 96% when whole face image was taken into consideration. A review on application of Gabor filters in different face recognition approach can be found in [96].

3.5 Feature Selection

After detection of fiducial points or face key points, the next step is to use the position or texture information from their location for building descriptors of face image to be recognized. The definition of face feature seems to have changed over period of time in computer vision community. Early research regarded face parts such as eyes, nose, mouth and so on as features as in [97]. As reported in [98], facial feature detection is mostly detecting facial features eyes, nose, mouth and face outline. The definition of face features seems to have broadened around 2005. In [99], face feature is defined as either region, key point(landmark) or contour. It is further reported in the paper that key point (or landmark) provides more accurate and consistent representation for face alignment compared to region based features. The landmark points are of special importance as some points are more discriminative than the others. The landmark localization theory is already explained in section 3.3. Gabor filter of different wavelengths and

frequency is applied on those landmark points. From the Gabor filter output, different feature extraction techniques can be applied. Gabor feature vector can be constructed [100]:

- Using Gabor Magnitude [101]
- Using Gabor phase [102]
- Using the real component only [103]

Gabor filters using magnitude response performs better than one using phase response [104]. Thus, Gabor magnitude is used to construct face feature vector in this thesis.

3.6 Support Vector Machine

Classification means to classify a new input/test data into one of the classes trained using training data. In face recognition, each individual is considered as a separate class and thus classifier classifies the test image into one of the classes defined during training. To classify the input image into one of the classes machine learning algorithm can be used. Machine learning algorithm learns a structure from a data set [105].

If $D=((x_1,y_1),(x_2,y_2),\ldots(x_n,y_n))$ be a sequence and if input 'x' and output 'y' function are used by machine learning algorithm to learn a pattern in a sequence the machine learning algorithm returns an approximate output y in response to an arbitrary input x [106]. Some of the machine learning algorithms are as follows: Decision tree learning, Artificial Neural Network, Support Vector Machines, Clustering, Bayesian Network, Genetic Algorithm. Support vector machine has shown a competitive recognition rate along with other machine learning algorithm such as Neural Network and Fuzzy Systems [107].

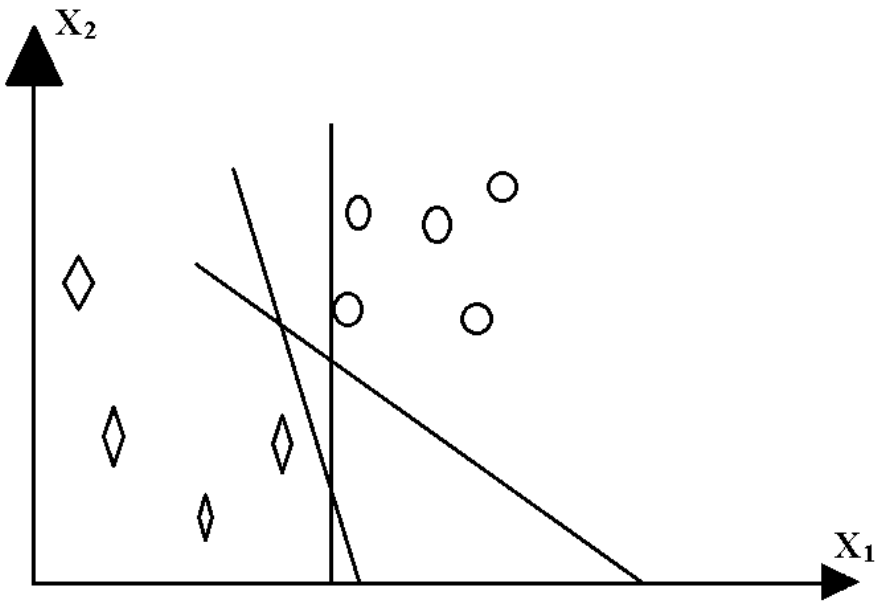


Figure 3.8: Different lines separating two sets of data points.

It was shown in [108] that despite the simple construction, SVM performed good in comparison to other machine learning algorithms such as two-layer Neural Network and Decision tree. SVM maps the input vector *x* into a high dimensional feature space *H* and construct an optimal hyperplane separating the features in that space [109]. The hyperplane known as Optimum Separating Hyperplane (OSH) minimizes misclassification rate not only in training examples but also in unseen samples of test set [108]. The OSH does so by maximizing the difference between positive and negative samples.

Let the training data $\{x_1,x_2,x_3,\ldots, x_n\}$ be labelled belonging to either positive or negative sample y={-1,1} [110]. From Figure 3.8, it can be seen that there could be different lines (hyperplane) separating one class from the other. The best line (hyperplane) is the one that maximizes the distance between two sets of data. The points which lies on this hyperplane satisfies [110]:

$$w.x+b=0 \tag{3.24}$$

Where, w is the normal vector to the hyperplane and $b/\|w\|$ is the offset of the hyperplane from the origin [110]. Any point lying on Hyperplanes H_1 and H_2 in Figure 3.9, i.e., the dark circle and dark diamond are called the support vectors, which changes the position of hyperplane dividing two sets of data [111] and distance between these set of hyperplanes (H_1 and H_2) is the margin $2/\|w\|$. To prevent data from falling in between the margin, the following constraint is added:

$$x_1 \cdot w + b \geq 1 \quad for \quad y_i = +1 \tag{3.25}$$

$$x_1 \cdot w + b \leq 1 \quad for \quad y_i = -1 \tag{3.26}$$

These set of equations can be further combined to:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i \tag{3.27}$$

For linearly separable case, SVM algorithm looks for the separating hyperplane which maximizes the margin [110]. Thus, SVM algorithm can be summarized as: finding *w* and *b* by minimizing $\|w\|^2$ subject to constraint Equation 3.27. For additional details please refer to [110]. During, the test phase, the test sample is determined to lie on one of the side of hyperplane H_1 or H_2 by using Equation 3.24 which gives its corresponding class [110].

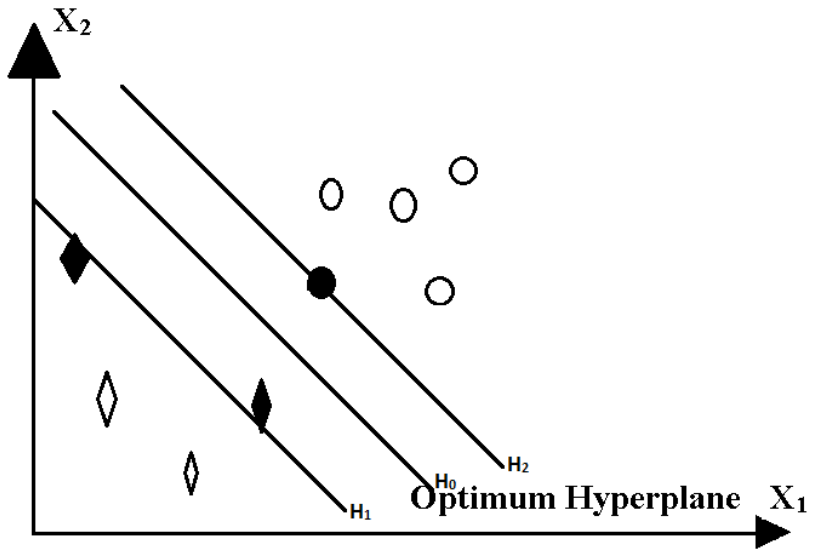


Figure 3.9: Optimum Hyperplane separating two data points (reproduced from [112]).

Although, SVM was initially used as binary classifier, it has proved to be good for multi-class classification as well. Binary SVM classifiers can be combined to solve a multi-class classification problem. Popular method for using the combination are one-versus-all (OVA) method, one-versus-one (OVO) method and error correcting methods [113].

If there are more than two classes, then the classification is done in higher dimensional space. Error correcting output code (ECOC) is used with application of binary classifier to solve multi-class classification approach in [114]. In multi-class classification by ECOC method, the problem of classification is broken down into coding and decoding scheme [114]. The coding step can be OVA or OVO [112]. In training step of OVA, *n* different bi-partitions (group of classes) are formed are formed and *n* binary problem classifier is trained [113]. In OVA training step, a code word of length *n* is generated for each class. Each bit of code is thus the response of one of the binary classifier. If the white region in Figure 3.10 is coded by +1 and black region by -1 then the class 2 is represented by the code word {-1,1,-1,-1} [114].

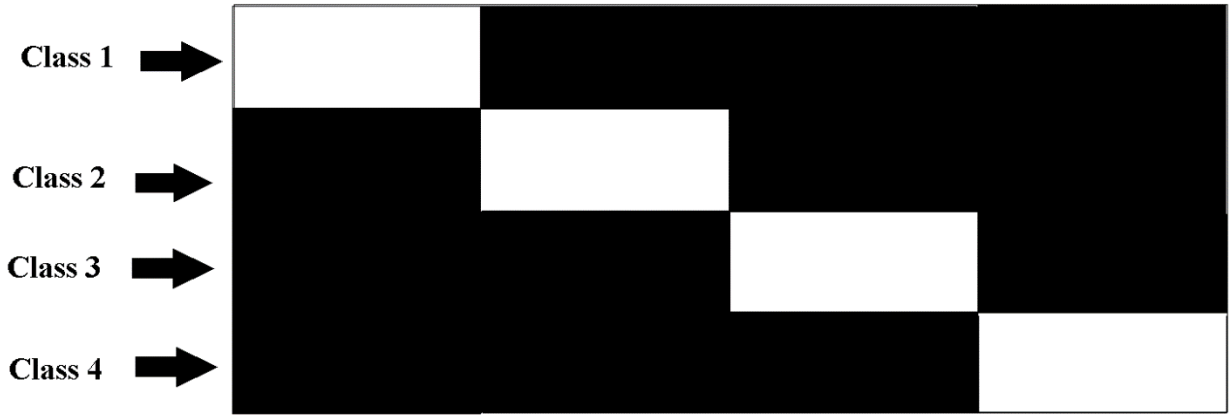


Figure 3.10: Multi-Class Classification using binary technique (reproduced from [114]).

In OVO coding scheme, *n(n-1)/2* binary classifiers are trained [112]. Thus, if there is a four-class classification problem, there are six binary classifiers. For each binary classifier, one with value +1 is positive sample, one with -1 is negative sample, and classes with 0 are not considered while training. The columns in Table 3.1 represents six classifiers for four class classification.

Table 3.1: One Vs. One Coding Matrix(reproduced from [112]).

Class 1	1	1	1	0	0	0
Class 2	-1	0	0	1	1	0
Class 3	0	-1	0	-1	0	1
Class 4	0	0	-1	0	-1	-1

In the decoding step, the output of these binary classifiers are grouped into a vector $V=\{B_1, B_2, B_3, B_4, B_5, B_6\}$ and the vector is compared with codeword of each class to assign a class label to the input sample. For details on decoding step, please refer to [115].

3.7 Different Human Detection Method

As head is the most discriminative part of human body, human detection is achieved by detecting human. In many human detection method, emphasis is given on background subtraction to limit the search area. In [32], the shape and temporal information is used as clue for human detection.

1. The method starts with moving object detection.

For moving object detection, background model is used to measure a pixel's probability of belonging to background $P_{BK}(x,y)$ based on mean and variance of static background images and Gaussian function. For the three-color channels, red '*r*', green '*g*' and blue '*b*', the mean $\mu_c(x,y)$ and variance $\sigma_c^{\,2}(x,y)$ is calculated based on the '*no*' number of static frames of background image. Here, $c \in [r,g,b]$ and $1 \le n \le no$.

$$\mu_c(x,y) = \frac{1}{no}\sum_{n=1}^{no} c_{Bn}(x,y) \tag{3.28}$$

$$\sigma^2{}_c(x,y) = \frac{1}{no}\sum_{n=1}^{no}\left[c_{Bn}(x,y) - \mu_c(x,y)\right]^2 \tag{3.29}$$

Then the probability vector $P_{BK}(x,y)$ is calculated as,

$$P_{BK}(x,y) = \left[\, p_{BKr}(x,y),\, p_{BKg}(x,y),\, p_{BKb}(x,y)\right]^T \tag{3.30}$$

Where, $p_{BGc}(x,y) = \dfrac{1}{\sqrt{2\Pi\sigma_c}}\exp\left(-\dfrac{\left[c(x,y)-\mu_c(x,y)\right]^2}{2\sigma^2{}_c}\right)$ (3.31)

2. After detection of moving object, edge detection is performed on moving foreground.

After detection of moving foreground, Canny edge detection is applied for further processing. As search area is already minimized by moving foreground object detection, canny edge detection is done only on moving object.

3. Connected edges are searched and grouped into different quadrants.

After edge detection, connected edges are grouped taking eight neighboring pixels into consideration. The connected edges are grouped into groups of ten pixels and stored for further processing.

4. Arcs from different quadrants are combined and ellipse is fitted.

From arc from different quadrants, the arcs whose lies in top most part of the moving object is selected in the beginning of ellipse fitting process.

5. Finally, head verification is performed.

The fitted ellipse is assumed to represent human head if the following three criteria are meet.

i) The fitted ellipse lies totally on the segmented moving foreground.

ii) The ratio of long axis and short axis of ellipse falls between 1.0 and 2.0.

iii) The center of ellipse should be located on upper central part of moving foreground object.

The drawback of this system is that human needs to be moving to be detected and multi-person cannot be detected. A similar method of using background subtraction and using color information of face is presented in [30].

In reference [30], motion detection is used as primary step for human detection. The human detection and tracking method consist of following steps:

- Motion Detection: The moving object is determined by taking difference from successive frames. Further, the image is thresholded so that the output of this stage is a moving binary object.
- Head Localization: To localize the head, the uppermost moving pixel is taken and other pixels are searched by considering a circle passing through these points using Hough Transform.
- Face detection: A statistical classifier is trained considering different skin color belonging to different race of people. In the method, skin color detection is applied only to the head region determined in the above step.
- Tracking: It is reported that face color provides important cue for tracking and controlling the tilt angle and pan angle. The mean value of the segmented face colored pixel is taken as the center of face while tracking.

It is reported in the paper that for slow moving persons such as people watching TV or washing dishes, the method can locate them with less than 10 cm error.

In reference [23], gradient and color module are used together for human detection. It is reported that since these two methods are almost complimentary, they serve to complement each other. In the paper, head is modeled as an ellipse of fixed ratio of 1.2 and $\sigma_{\min}$ is the length of minor axis. The tracking task is find better estimate of head location using gradient and color information. The goodness of fit is obtained by:

$$S^* = \arg\max_{S_i \in S} \left\{ \overline{\varphi_g}(S_i) + \overline{\varphi_c}(S_i) \right\} \tag{3.32}$$

Where, $\overline{\varphi_g}(S_i)$ and $\overline{\varphi_c}(S_i)$ are the matching scores based on intensity gradients and local histogram respectively. The color score is obtained as follows:

$$\varphi_c(s) = \frac{\sum_{i=1}^{No} \min(I_{HS}(i), M_H(i))}{\sum_{i=1}^{No} I_{HS}(i)} \tag{3.33}$$

Where, $I_{HS}(i)$ and $M_H(i)$ are the number of pixels in the ith bin of image and model histograms respectively and N_o is the number of bins. To compute the gradient magnitude around the ellipse perimeter, following formula is used:

$$\varphi_g(s) = \frac{1}{N} \sum_{i=1}^{N} \left| n(j).g_s(j) \right| \tag{3.34}$$

Where, $g_s(j)$ is the intensity gradient at the perimeter of ellipse pixel *j*, *n(j)* is the unit normal vector to the pixel *j*, *N* is the number of pixel on perimeter of ellipse and (.) denotes the dot product. To add gradient score '$\varphi_g(s)$' to color score '$\varphi_c(s)$' both color score and gradient score are converted to percentage ($\overline{\varphi_g}(s)$, $\overline{\varphi_c}(s)$) by subtracting the score from minimum and dividing by the range. The advantage of this method was that it could detect human even when the head had large variation in angle. The limitation of this method as reported in the paper is that lighting conditions effects the head tracking performance.

3.8 Different Face Recognition method

The use of Gabor filter has dominated the face recognition research. In [116], first log-polar transform (LPT) is applied on the input image. After application of LPT, Gabor filter is applied on the fiducial points on log-polar transformed coordinates of the fiducial points. It is shown that the method performs better than Elastic bunch graph matching method. Gabor filter is applied on 16 different facial parts. The co-ordinates of the fiducial points are selected manually. If *I(x,y)* is the coordinated of the image in Cartesian coordinate then the LPT image is given by:

$$I*(\varsigma,\phi) = L\left\{I(x,y),(x_{lo},y_{lo})\right\} \tag{3.35}$$

Where, $\varsigma = M\log(r)$ (3.36)

$$r = \sqrt{(x-x_{lo})^2 + (y-y_{lo})^2} \tag{3.37}$$

$$\phi = \tan^{-1}\left(\frac{y-y_{lo}}{x-x_{lo}}\right) \tag{3.38}$$

The parameter *M* is reported to be determined from the height of the image. For implementation of Gabor filter, five frequencies and eight orientations are used. Two compare similarity between different jets G and G', the similarity function used is follows:

$$S_{jet}(G,G') = \frac{\sum_{G=1}^{G_f} a_G a_G{}'}{\sqrt{\sum_{G=1}^{G_f} a_G{}^2 \sum_{G=1}^{G_f} a_G{}'^2}} \tag{3.39}$$

Where, a_G is the result of convolution between the real and imaginary part of Gabor filter and image. Now, to compare all the jets of a query image *Q* with database image *D*, the following similarity function is used.

$$S_{total}(G^Q,G^D) = \frac{1}{N}\sum_{n=1}^{N} S_{jet}(G_n{}^Q, G_n{}^D) \tag{3.40}$$

Where, *N* is the total number of Gabor jets.

It is mentioned that in this research, the fiducial points were located manually. In other research [117], the fiducial points has been located automatically. In [117] it is shown that the combination of Gabor coefficients on fiducial points and Geometric distance produces a good result. The face region in their process is selected using skin color thresholding using in YCbCr color space. After that for localizing fiducial points, again color information is used. The test was conducted on FERET database [118] for five combination of face samples. The geometric distance between various facial parts is calculated as:

$D_{eyetoeye}$ is distance between center of two eyes

D_{eye}: width of eye

$D_{betweeneye}$: shortest distance between two eyes

D_{nose}: width of nose

D_{eye_nose}: distance between center of eyes and center of nose:

D_{mouth} : width of mouth

D_{nose_mouth}: Distance between center of nose and center of mouth

After calculation of geometric distance, 1 X 7 feature vector V is constructed using geometric distance as follows:

V=[$D_{eyetoeye}$; D_{eye} ; $D_{betweeneye}$; D_{nose_width} ; D_{eye_nose}; D_{mouth_width}; D_{nose_mouth}]

Additionally, Gabor features are obtained on ten points; three on eyes, two on extremes of nose and two on the corners of mouth. The orientation of Gabor filter and wavelength is chosen as the parameters chosen in [41].

i.e Orientation $\theta = \left\{0, \frac{\Pi}{8}, \frac{2\Pi}{8}, \frac{3\Pi}{8}, \frac{4\Pi}{8}, \frac{5\Pi}{8}, \frac{6\Pi}{8}, \frac{7\Pi}{8}\right\}$ and

wavelength $\lambda = \left\{4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\right\}$

While the use of color information is a very good start for automatic landmark localization, it can be inaccurate sometimes depending upon the lighting direction, and presence of beard, moustache and so on. Additionally, presence of light or dark color eyelashes may affect the localization of eyes. The result is shown for one orientation of Gabor filter and increases by one up to 5 orientations. Also, the result is shown based on dividing the set on 60% training and 40% testing, 50%-50% and 30% - 70%. The result is tabulated below.

Table 3.2: Face Recognition Rate.

Variables	60%-40%	50%-50%	30%-70%
5 Gabor wavelets	98.90%	98.20%	96.50%
10 Gabor wavelets	99.30%	99%	98.30%
15 Gabor wavelets	99.40%	98.80%	98.60%
20 Gabor wavelets	99.30%	99.40%	99.20%
25 Gabor wavelets	99.70%	99.70%	100%

In another research [119], face recognition using Gabor filter and Local Binary Pattern (LBP) is presented. Local Binary pattern encodes small texture details and Gabor wavelets represent multi-scale representation of an image. As both LBP and Gabor features are high dimensional, PCA is applied before fusing these features. The Local binary pattern gives an eight-bit code comparing the gray level of central image pixel under consideration with 8 neighborhood image pixel. The binary code is generated as shown in Figure 3.11.

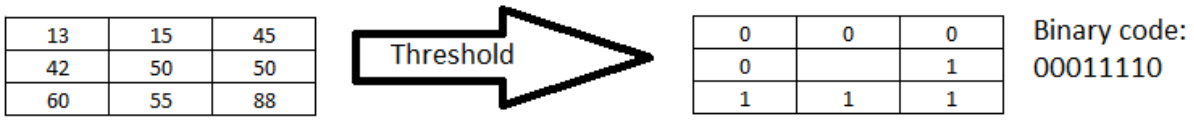


Figure 3.11: Application of Local Binary Pattern (reproduced from [119]).

Gabor filter is applied on the whole image face image and PCA is applied for dimensionality reduction. If a face is represented by *N*-dimensional vectors then PCA seeks *M* orthogonal variation that captures maximum variation in face set [119]. These directions of maximum variation can be encoded into a $M \times N$ matrix *U*. Let, x_g and x_l be the Gabor feature and LBP features of the face image respectively and let μ_g and μ_l be their mean. If $y_g = U_g^T(x_g - \mu_g)$ and $y_l = U_l^T(x_l - \mu_l)$ are the corresponding centered and PCA reduced vector, then the combined feature *z* is the 'z-score' normalized combination

$$z = (y_g / \sigma_g, y_l / \sigma_l)^T \tag{3.41}$$

Where, σ_g and σ_l are standard deviation of y_g and y_l respectively. From the combined feature space *z*, kernalized version of LDA is used to extract unique features. The combined feature vector z_{query} is projected into the optimal plane space as:

$$\Psi_{query} = P^T \Phi(z_{query}) \tag{3.42}$$

Where, *P* is the Projection matrix and $\phi(z_{query})$ is the query image sample mapped in kernel induced feature space. For details on derivation of these two terms, please refer to [119]. The feature vector is classified using nearest neighborhood rule and cosine distance.

$$d_c(\Psi_{query}, \Psi_{database}) = -\frac{\Psi^T_{query} \Psi_{database}}{\|\Psi_{query}\| \|\Psi_{database}\|} \tag{3.43}$$

Where, $\Psi_{database}$ is a face template in the stored database.

3.9 Summary

In this chapter, some of the relevant theories and theory for the method used to detect human head and recognize human face was presented. More specifically, theory of curvature, approximation of curvature of a curve and mathematical formula for estimation of curvature of a curve in 2D image was presented as it is the main idea for detection of human using shape information. Further, theory on automatic landmark localization, feature extraction and classification was presented. Additionally, few human detection method and face recognition method was also presented. In the next chapter, the proposed method to detect human will be presented in detail.

Chapter 4

PROPOSED METHOD OF HUMAN DETECTION AND RESULT

4.1 Introduction

For HRI, the vision system attached to the robot should be able to identify human in its vicinity. Human detection system attached to a robot should be able identify human presence in its surrounding and localize them. In this chapter, the method used to detect human in this thesis is elaborated in detail.

4.2 Problem Definition

The problem in human detection is to detect and localize person so that the interaction with human can be started. Human detection by computer vision system can be very different from human visual system. However, shape and color information is used by both human and computers to detect human. Next, various steps carried out to detect human in indoor environment will be discussed.

4.3 Pre-Processing

Before proposed curvature method is applied in the input image, some pre-processing steps can be taken which can eliminate other edges which are not likely to represent human head. By eliminating those curves, the calculation in the succeeding steps can be reduced. In the proposed method, linear or almost linear lines are removed by applying angle criteria and line fitting after detecting the edge from input image. Edge detection and process of removal of linear segment is described next.

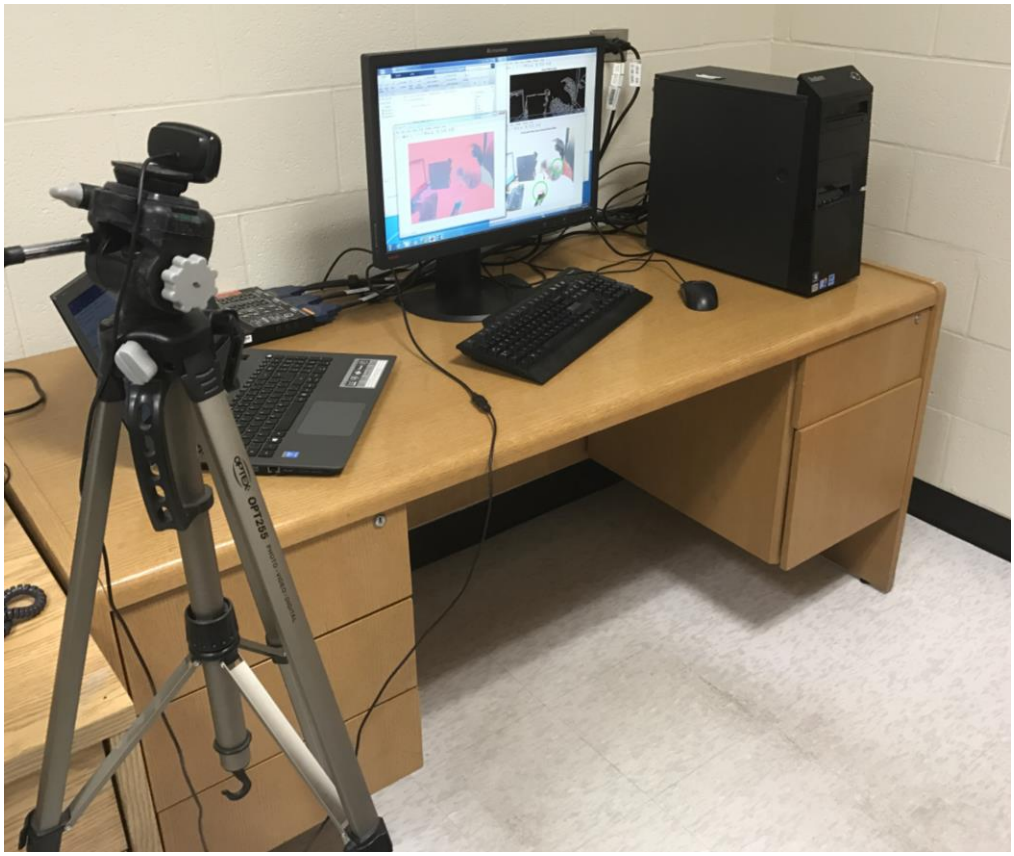


Figure 4.1: Camera Setup.

4.3.1 Edge Detection

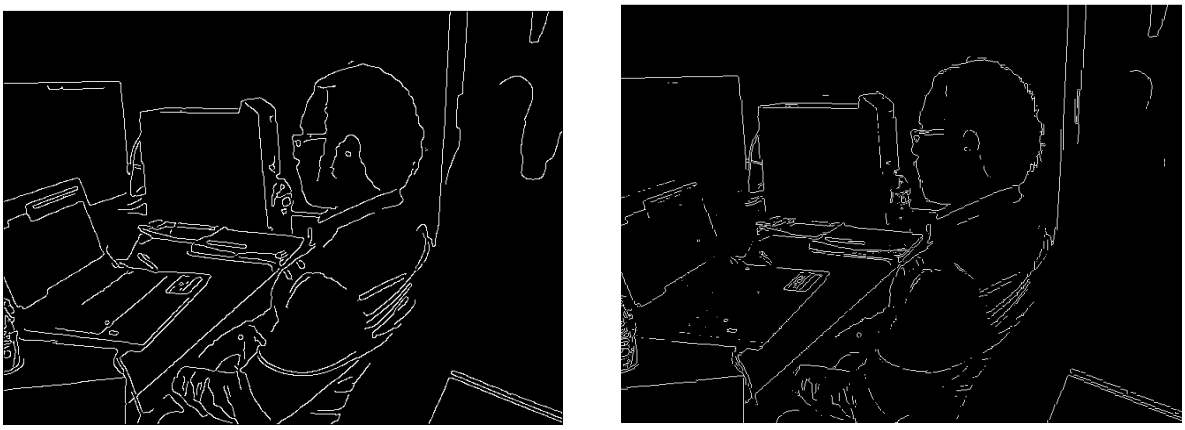
Basically, 2-D shape analysis is the fundamental step to recognize and locate object in digital images [120]. In this thesis, a Logitech c615 camera as shown in Figure 4.1 is used. The camera is fixed but it can also be fixed on the top of a mobile robot. To compare the shape of object in the image plane with that of human head, the edge or the contour of an object should be compared with human head curve. Edge refers to the places in the image where there is sudden change in brightness value. To detect an edge, an edge detector algorithm is applied. An edge detector accepts digital images as input and produces an edge map as output [121]. Thus, in this thesis, first, the input RGB image as shown in Figure 4.2(a) is converted to a grayscale image as shown in Figure 4.2(b). Then, a specific edge detector method is applied to detect the edges. Two

commonly used edge detection methods are: ***i***) Canny edge detection; ***ii***) Prewitt edge detection. Classical edge detectors are capable of detecting edges and their orientations, and are simple in computation [121].

(a) (b)

Figure 4.2: Input Image (a) RGB image (b) Grayscale image.

For edge detection, Canny edge detector is used as it is very accurate. There are other Edge detectors available in MATLAB such as Robert, Sobel and so on. Initially Prewitt edge detector was used, however it failed to capture head contour sometimes. As edge detection is first step, its output will affect the subsequent stages and thus result of human head detection algorithm. Canny edge detector is preferred in object detection algorithms as it gives a very detail and accurate representation of an image.



(a) (b)

Figure 4.3: Edge Detection (a) Canny Edge Detection (b) Prewitt Edge Detection.

Canny edge detector produces accurate output edges. Canny has presented three main criteria for edge detectors [122]; good detection, good localization and single response to a single edge and found that the optimal detector is the first derivative of the Gaussian [123]. Canny edge detector gives a very detailed and accurate edge as shown in Figure 4.3(a) compared to Prewitt Edge detection which is shown in Figure 4.3(b). As the head contour is one of the most important feature for human detection, Canny edge detection is used in this research to find the head contour from the edge image. To calculate Canny Edge, first a Gaussian mask is applied to an image to eliminate noise present in the image [124]. After that horizontal edge, E_X and vertical edge strength E_Y is calculated. The edge strength in these two directions give the total edge strength E.

$$E = |E_X| + |E_Y| \tag{4.1}$$

After calculating the total edge strength, edge direction is calculated as follows:

$$\theta_{edge} = \tan^{-1}\left(\frac{E_Y}{E_X}\right) \tag{4.2}$$

After finding edge direction, non-maximum suppression is used which assign a value of 0 to all pixels except the edge pixel. To localize head, other edges which don't represent human head curve should be omitted. These procedures are explained in the following sections.

4.3.2 Straight Line Deletion

Automatic detection of lines in an image is the fundamental step in many computer vision and processing tasks [125]. Hough transforms which was initially proposed to detect curvilinear structure is used in line detection as well. For Hough transform, exact shape of the object to be detected should specified which makes the process very difficult and even impractical for

implementation [126]. After finding the edge as shown in Figure 4.3, it can be noticed that there are many edges which doesn't represent human head. The edges which are very small; less than 5 pixels are removed as they obviously won't represent human head. One of the MATLAB programs [127] is used to find connected edges. As can be seen in Figure 4.3(a), there are many unwanted edges and elimination of those edges makes the method faster as working on fewer numbers of pixels lowers computational costs. In this thesis to eliminate the linear segments, each particular edge in the image is divided into number of smaller segments then two-point form of a line is applied to find the slope of various segments. The slope of a section of one particular line segment of each edge is calculated using two-point method, i.e., where (x_1, y_1) and (x_2, y_2) are the first and the last points of the selected segments respectively, and m is the slope of line. From the calculated slope, the angle made by two consecutive tangent lines on the segment can also be obtained as:

$$\theta = tan^{-1}(m) \quad . \tag{4.3}$$

Thus, the angle made by various segments and the difference of angle between consecutive segments is calculated as shown in Figure 4.4.

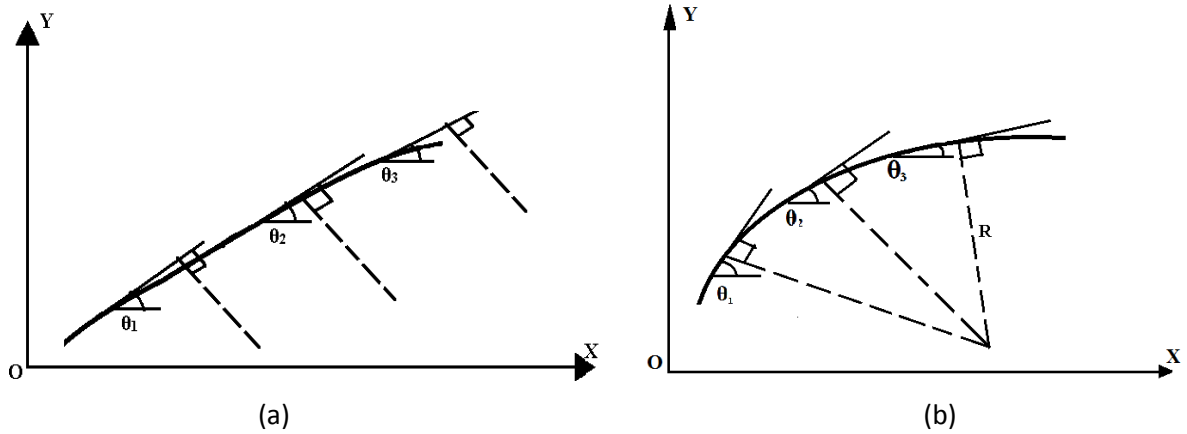


Figure 4.4: Application of angle thresholding: (a) Linear Segment (b) Circular Segment

In Figure 4.4(a), the difference between angles made by tangents of segments and horizontal axis are small. It can be also seen that the normal to the tangents are almost parallel indicating that those segments can be considered as linear segment. By choosing a suitable threshold angle ($\theta_{threshold}$), linear segments can be omitted. In Figure 4.4(b), the angle difference is greater than the threshold i.e. $\theta_2 - \theta_1 > \theta_{threshold}$ and normal drawn from different tangent intersect at a point. The point of intersection of the normal lines drawn from various segments gives the approximate center of the curve. Thus, segment shown in Figure 4.4(b) can be considered as a part of a curve with certain curvature and it can potentially represent a human head. Therefore, all segments which have angle difference greater than the threshold are retained for further investigation.

Figure 4.5: Straight line deletion using angle criteria.

After elimination of linear edges, Figure 4.5 is obtained. In Figure 4.5, there are certain linear segment which couldn't be eliminated from the preceding stage. While a larger threshold could eliminate more number of straight line segments, certain segment from head might also be

eliminated. Additionally, some objects which are linear appear as slanted lines in image plane and thus can't be eliminated in the first step. To prevent edge in head from being deleted and to delete other linear segments, an additional line fitting process is applied to remove linear segments.

To fit lines in various segment of Figure 4.5, Matlab function Polyfit is used. Polyfit(*x*,*y*,*n*) generates coefficients of polynomial $p(x)$ of order n that fits the points $P(x,y)$ in an edge denoted by white lines in Figure 4.5. For line fitting, polynomial $p(x)$ is of order 1.

$$p(x) = p_1(x) + p_2. \tag{4.4}$$

After coefficients p_1 and p_2 of the polynomial $p(x)$ is calculated, the vertical coordinate of line that best fits the given points is given by:

$$\hat{y} = p_1 x + p_2 \tag{4.5}$$

Then the error e, i.e. difference between actual value of y and predicted value of y, i.e. $\hat{y}$, is calculated as:

$$e = y - \hat{y} \tag{4.6}$$

And the total Root mean square (RMS) error for the fitted line is calculated as:

$$RMSE_{line} = \sum_{i=1}^{N} \sqrt{\frac{e_i^{\,2}}{N}} \tag{4.7}$$

Where, $i = 1, \ldots N$ is the total number of points on a segment. The blue lines in Figure 4.6 are the lines fitted to various edges remaining from the preceding stage. The RMS error of line fitting process is shown in the first pixel of every edge in Figure 4.6. It can be seen from Figure 4.6 that lines have smaller RMS error whereas, curves have larger RMS error. Please refer to

Figure 4.5 and Figure 4.7 for edge images; before and after applying this process. Thus, this criterion can eliminate linear segments which weren't eliminated from the preceding stage (angle difference criteria).

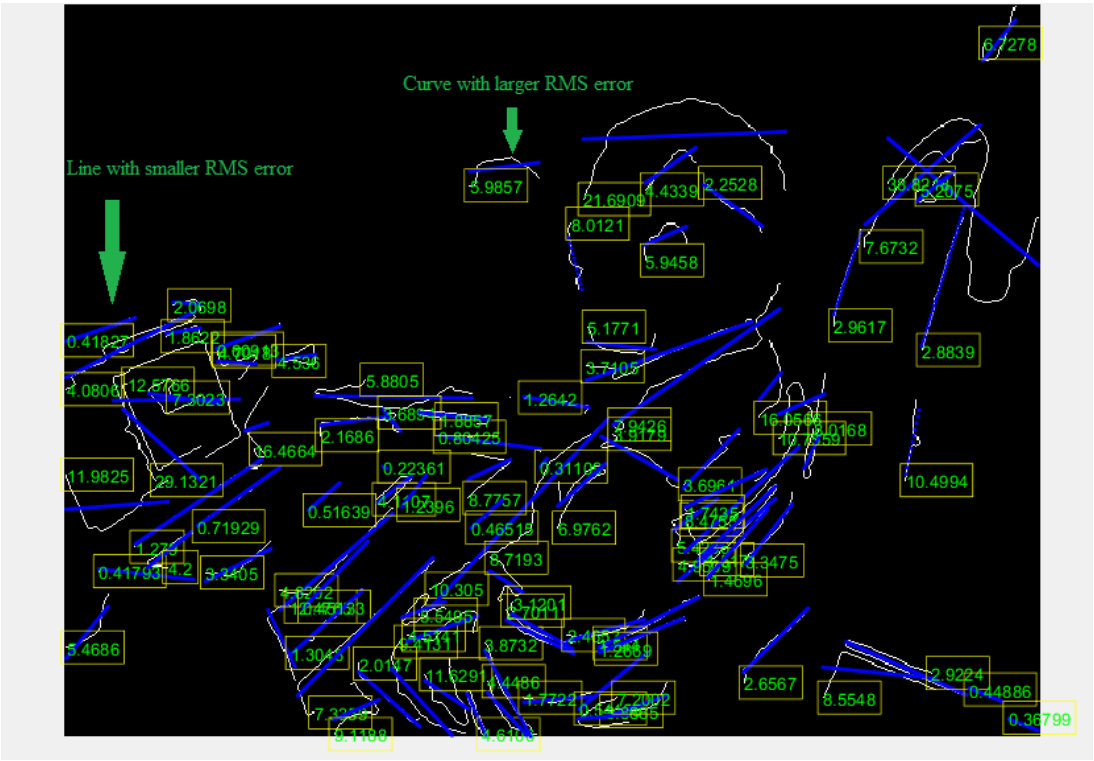


Figure 4.6: Fitting line to various curves.

For vertical lines, however, the independent value x in Equation 4.5 is constant thus dependent value $\hat{y}$ will also be constant. A separate criterion is thus needed to check vertical lines. To check whether a line is vertical, it is divided into two parts and angle between first point and middle point 'θ_1' and angle between middle point and last point 'θ_2' is calculated. If both the angles are greater than 85°, i.e. $\theta_1 > 85^o$ and $\theta_2 > 85^o$, the line is vertical and is thus eliminated. If the line is not vertical the process presented through Equations. (4.4-4.7) is applied to calculate RMS error. If RMS error of line fitting process is less than a threshold, it can be eliminated,

Figure 4.6. After elimination of lines, Figure 4.7 is obtained.

Figure 4.7: Edge Image obtained after using line fitting criteria.

From the curves, which are left out, the task is to find the curve having curvature close to that of human head, which is described next.

4.4. Curvature Calculation

In this thesis, before calculating curvature of human head, curvature of synthetic object and some real objects was calculated. In following paragraphs, curvature calculation of synthetic and real object is presented first and curvature calculation of human head is presented then after.

4.4.1. Curvature Calculation of different objects

The accuracy of calculating curvature using Equation 3.12, is initially evaluated using a synthetic image of circle of known radius (i.e., R =199 pixel) which is created in Paint software as shown

in Figure 4.8. To calculate the curvature, first, the synthetic image is processed using edge detection algorithm and then curvature of the detected edge is calculated using Equation 3.12, and the results are shown in Figure 4.9 and listed in Table 4.1.

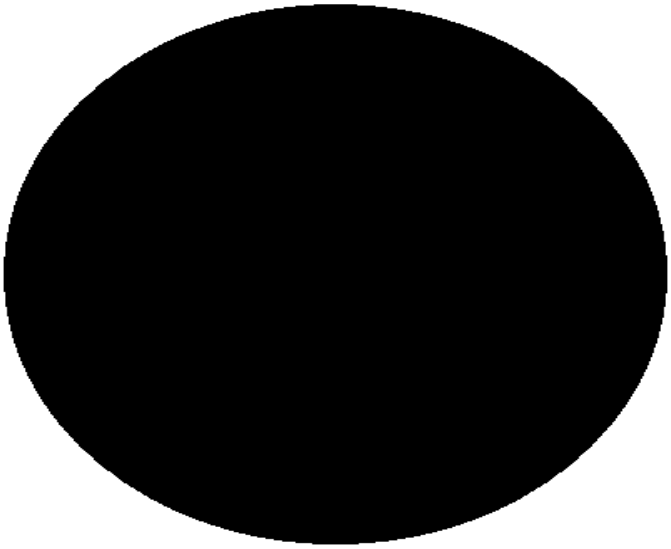


Figure 4.8: A synthetic circle created using Paint Software.

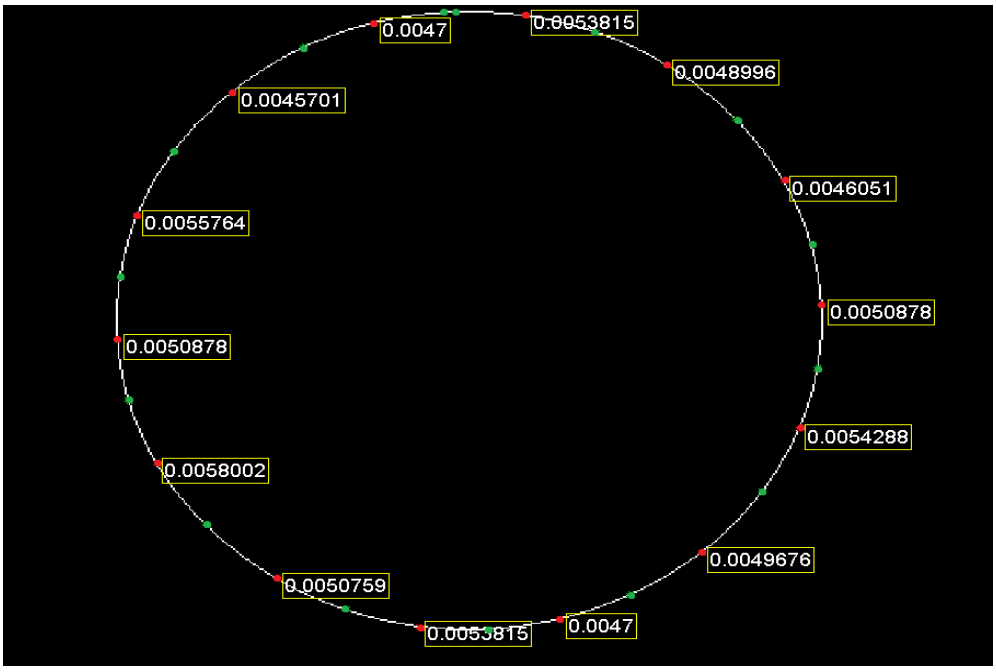


Figure 4.9: Curvature calculation of circle shown in Figure 4.8.

The circle is divided into the number of segments and red points denote the center of each segment enclosed by green points. The results shown in Figure 4.9 is calculated based on 80 pixels for each segment, and comparison of the results with the curvature calculated using $\kappa = 1/R$ is 0.005 pixel shows that the average error is 6.29%. This error percentage can be reduced by increasing number of pixels used to form each segment. It can be noted from Figure 4.9 that Equation 3.12 gives a good approximation for calculation of curvature of a curve in 2D digital image. Also, from Table 4.1, it can be seen that average absolute error on calculation of curvature using Equation 3.12 is 0.0003 pixels. Since, synthetic image gives the same curvature in every trial, it was not necessary to repeat the calculation.

Table 4.1: Curvature Calculation of a synthetic circle.

Segment	Calculated Curvature (pixel)	Real Curvature (pixel)	Absolute Error (pixel)	Error Percentage (%)
1	0.0054	0.005	0.0004	8
2	0.0049	0.005	0.0001	2
3	0.0046	0.005	0.0004	8
4	0.0051	0.005	0.0001	2
5	0.0054	0.005	0.0004	8
6	0.0050	0.005	0.0000	0
7	0.0047	0.005	0.0003	6
8	0.0054	0.005	0.0004	8
9	0.0051	0.005	0.0001	2
10	0.0058	0.005	0.0008	16
11	0.0051	0.005	0.0001	2

12	0.0056	0.005	0.0006	12
13	0.0046	0.005	0.0004	8
14	0.0047	0.005	0.0003	6
		Average	0.0003	6.29

At this point, the accuracy of curvature approximation using Equation 3.12, is checked on image of a circular object (flying disc) as shown in Figure 4.10.

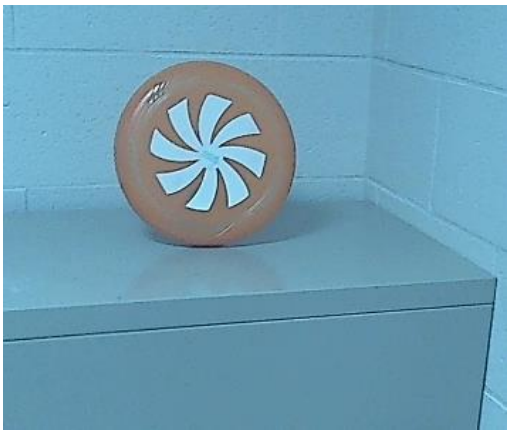
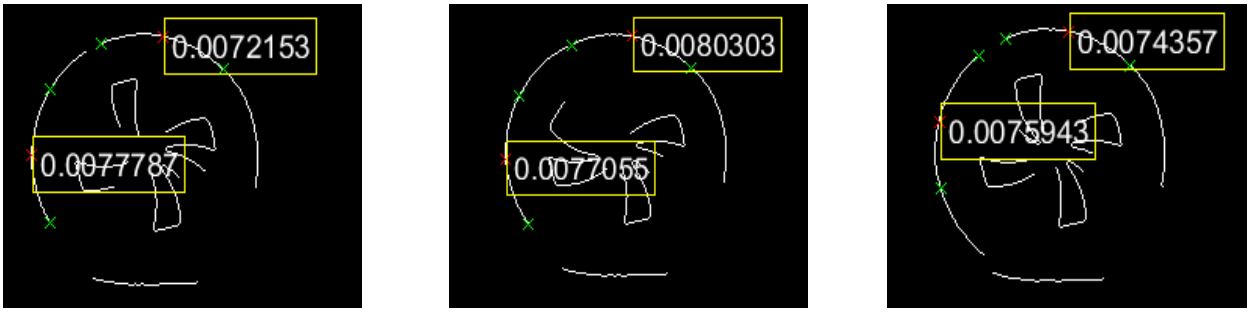


Figure 4.10: Flying Disc.

To calculate the curvature of a real object, first the image calibration factor is obtained using another object with known width of 239 mm which is placed at 1160 mm from the camera. Its width in image plane is calculated to be 132 pixels. Thus, pixel to mm conversion factor at that location is found to be 239/132=1.811. Next, a flying disc with a diameter of 250 mm as shown in Figure 4.10 is taken. Then, using Equation 3.1, i.e., $\kappa = 1/R$, its curvature is found to be 0.008 mm. Here, the number of pixel used to form a segment is 80. This experiment is repeated three times as shown in Figure 4.11 and the curvature is calculated in mm. For each trial average of curvature calculation is obtained and the error is shown in Table 4.2.



(a) (b) (c)

Figure 4.11: Curvature Calculation of flying disc: (a) Trial 1 (b) Trial (2) (c) Trial 3.

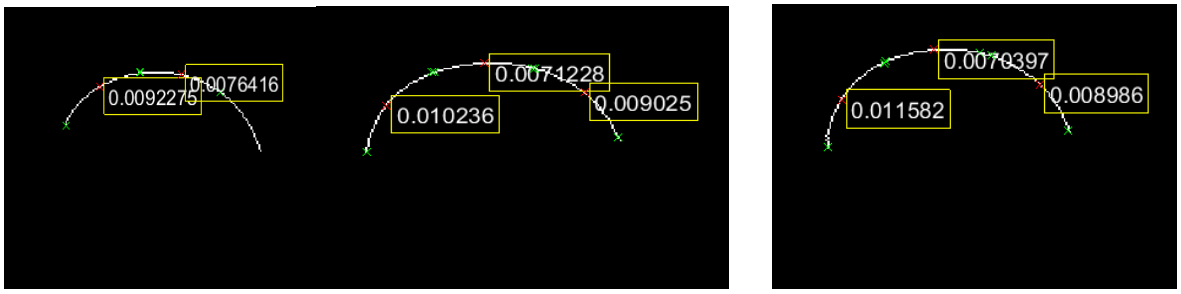
Table 4.2: Curvature Calculation of the flying disc.

Trial	Measured curvature (κ_a)	Calculated curvature (κ)	Absolute error	Error percentage (%)	Implied diameter $\left(D = \frac{2}{\kappa}\right)$
1	0.008 mm	0.0075 mm	0.0005 mm	6.25	266.67 mm
2	0.008 mm	0.0079 mm	0.0001 mm	1.25	253.16 mm
3	0.008 mm	0.0075 mm	0.0005 mm	6.25	266.67 mm

The average calculated curvature using Equation 3.12 is found to be 0.00750 mm, 0.00787mm and 0.00752 mm respectively. It can be noted that calculated curvature are close to the measured value of 0.0008 mm. The bottom most part in Figure 4.11 appears linear instead of circular as it is edge formed by the shadow as can be seen in Figure 4.10. The error in curvature calculation can be due to measurement error in calculation of the dimension of the object or due to digitization of the image.

Figure 4.12: A hard hat.

Next a hard hat of 250 mm length, used commonly in industries and laboratory, is taken as shown in Figure 4.12. Its length is 250 mm and again using Equation 3.1, i.e. $\kappa = 1/R$, its curvature is calculated to be 0.008 mm. The curvature calculation was again repeated three times as shown in Figure 4.13. and in this case the number of pixel used to calculate curvature was 50 pixels instead of 90 pixels. The number of pixel is taken to be 50 because curvature of human head is also calculated by forming segments of 50 pixels.



(a) (b) (c)

Figure 4.13: Curvature Calculation of a hard hat: (a) Trial 1 (b) Trial 2 (c) Trial 3.

Table 4.3: Curvature Calculation of hard hat.

Trial	Measured Curvature (κ_a)	Calculated Curvature (κ)	Absolute Error	Error (%)	Implied Length ($L = 2/\kappa$)
1	0.008 mm	0.0084 mm	0.0004 mm	5.43	237.12 mm
2	0.008 mm	0.0088 mm	0.0008 mm	9.93	227.41 mm
3	0.008 mm	0.0092 mm	0.0012 mm	15.03	217.33 mm

For each trial, average of curvature calculation is obtained and the error is shown in Table 4.3. The average values of curvature were 0.0084 mm, 0.0088 mm and 0.0092 mm compared to measured value of 0.008 mm. As number of pixel taken to form a segment is reduced, here the average error increases slightly. From above examples, it is seen that by calculating curvature objects with known curvature can be detected. Human head has circular shape and it will have

certain curvature. By using average human head length and width [128] as shown in Figure 4.14, the average human curvature facing camera from different angles can be calculated. The average human head dimension is as shown in Figure 4.14.

Table 4.4: Average Human Head dimension.

Subject	Head width (W)	Head length (L)
Human Average	165.4 mm	205.4 mm
Average Human Curvature	0.0121 mm	0.0097 mm
Subject used in this thesis	158 mm	195 mm
Curvature	0.0127 mm	0.0103 mm

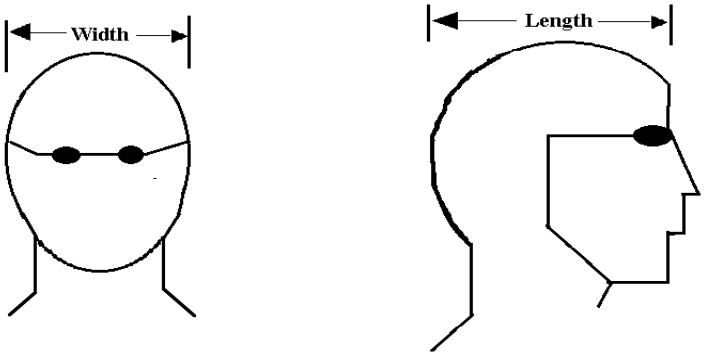


Figure 4.14: Width and Length of Human Head (reproduced from [129]).

4.4.2. Curvature Calculation of Human Head

The application of curvature calculation method for an input image is as shown in Figure 4.15. The segment of curves which are within the threshold are shown in green rectangle. While the rest, shown in red rectangle, are beyond the threshold and thus are eliminated.

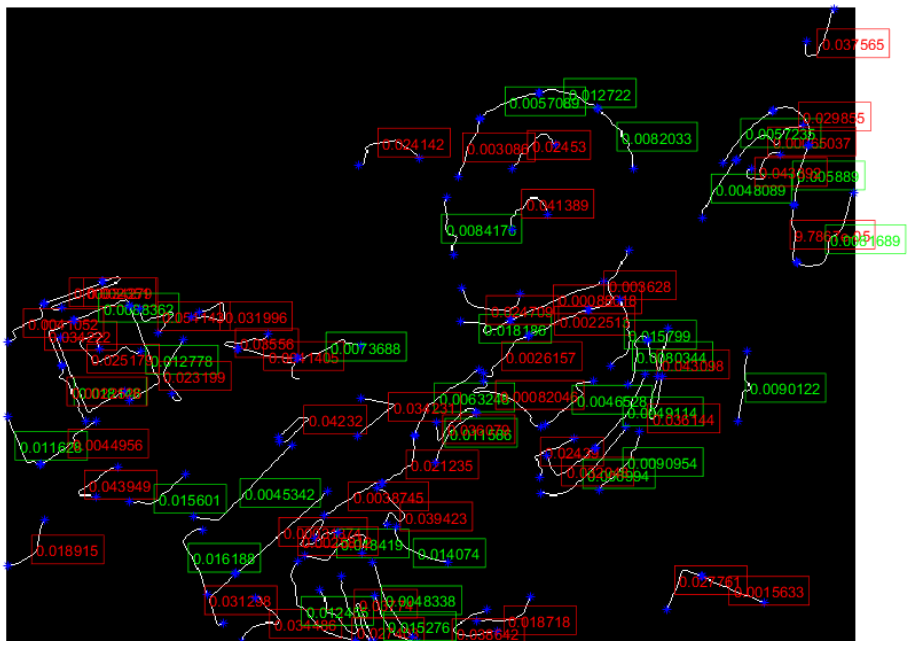


Figure 4.15: Calculation of head curvature.

In this thesis, after finding the curvature of different edges identified, only those curves which has curvature value close to that of reference/average (i.e., 0.0115 -/+ 0.007 mm) human head is retained and the rest is eliminated. After application of curvature method, the results shown in Figure 4.16 is obtained, which shows not all curves are eliminated.

Figure 4.16: Application of Curvature Criteria.

The length and breadth of different segments can be measured to retain curves and eliminate linear segments. While curves have smaller major axis length to minor axis length ratio, linear segments have higher ratio. It can be also noted that the aspect ratio (length of major axis to length of minor axis of an ellipse enclosing human head) is taken to be 1.2 in [23] and values between 1.0 and 2.0 in [32]. While these thresholds can help to segment perfectly elliptical shapes, in this thesis to avoid elimination of potential human head curve segments a more conservative approach, with higher threshold (i.e., 9), is chosen to retain other curves because some curves representing human head might be broken into several small segments. Segments having major axis length to minor axis length ratio more than the set threshold, i.e. 9, will be discarded. However, if a connected component has a major axis length of 48 pixels and minor axis length of 6 pixels, its ratio will be 8 and thus will be retained. The threshold ratio of 9 is determined according to the observation in this experiment and thus the ratio may be different in other applications. After application of length to breadth ratio, Figure 4.17 is obtained, which shows fewer curve are remaining.

Figure 4.17: Length to breadth thresholded image.

4.5. Circle fitting

Circle fitting can be done on various curves, some curve will have a very small radius and some curve will have a radius big enough to be human head radius. Thus, by selecting only those curves whose radius is bigger than a reference value, all other curves can be neglected on further processing. The least square fit of circles and circular arc is based on minimizing the mean square distance $1/N \sum_{i=1}^{N} d_i^2$ from the circle to the data points (x_i,y_i) [130], where N is the number of data points and d_i is the Euclidian distance. If the points on remaining segment of the curve circle satisfies the following equation:

$$(x-a)^2 + (y-b)^2 = R^2. \tag{4.8}$$

where (a,b) is the center and R is the radius. Then the distance d_i (i.e., error of data from desired circle) can be calculated as follows:

$$d_i = \left(\sqrt{(x_i - a)^2 + (y_i - b)^2} \right) - R \tag{4.9}$$

By minimizing the objective function $f = 1/N \sum_{i=1}^{N} d_i^2$ (i.e., mean square errors), which is a nonlinear least square problem, the three parameters of the circle; center and radius of the circle can be calculated. For details, the reader is referred to reference [130]. After finding R, it should be compared with reference/average human head diameter (approximately 165mm [128]), if it is in the range of human head, then the given segment can be considered as human head contour. Since it is assumed that the approximate distance of human from camera is known, the radius of circle encircling the human head can be approximated. In this research, only those circles having radius more than 40 pixels is considered for further steps. Thus, application of circle fitting

process in Figure 4.18 is shown only for those curves whose radius is more than 40 pixels. The values in green boxes are the RMS error of each curves. The RMS error calculation is described next.

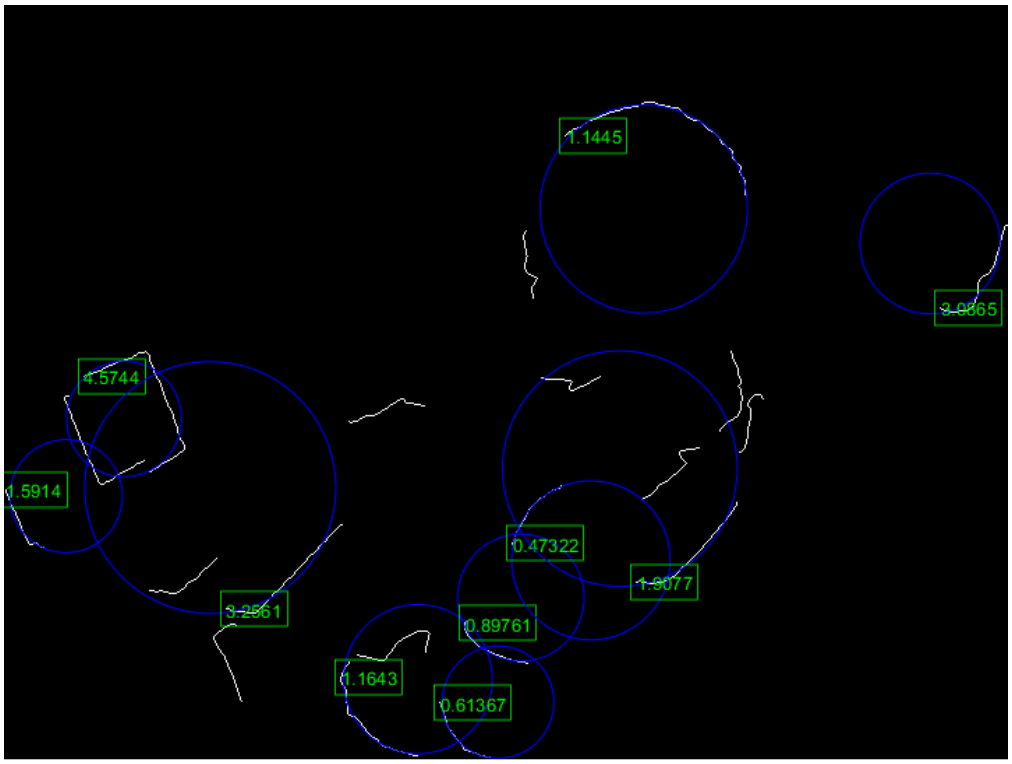


Figure 4.18: Circle fitting on a curve.

4.6 Root Mean Square Error Calculation

Additionally, after circle fitting process the closeness of different curves to the fitted circle can be calculated by root mean square error calculation. For a point $P(x, y)$ on the curve, its deviation (e) from the fitted circle with radius (R) and center (x_c, y_c) is calculated as:

$$e = R - \sqrt{(x - x_c)^2 + (y - y_c)^2}. \tag{4.10}$$

While circular curves will have lesser deviation from fitted circle, other curves will deviate more and thus will have relatively larger root mean square error. The root mean square error (E_{RMS}) for a curve is calculated as follows:

$$E_{RMS} = \sum_{i=1}^{N} \sqrt{\frac{e_i^2}{N}}. \tag{4.11}$$

Where, $i = 1, 2, \ldots N$ is the total number of pixels in a curve.

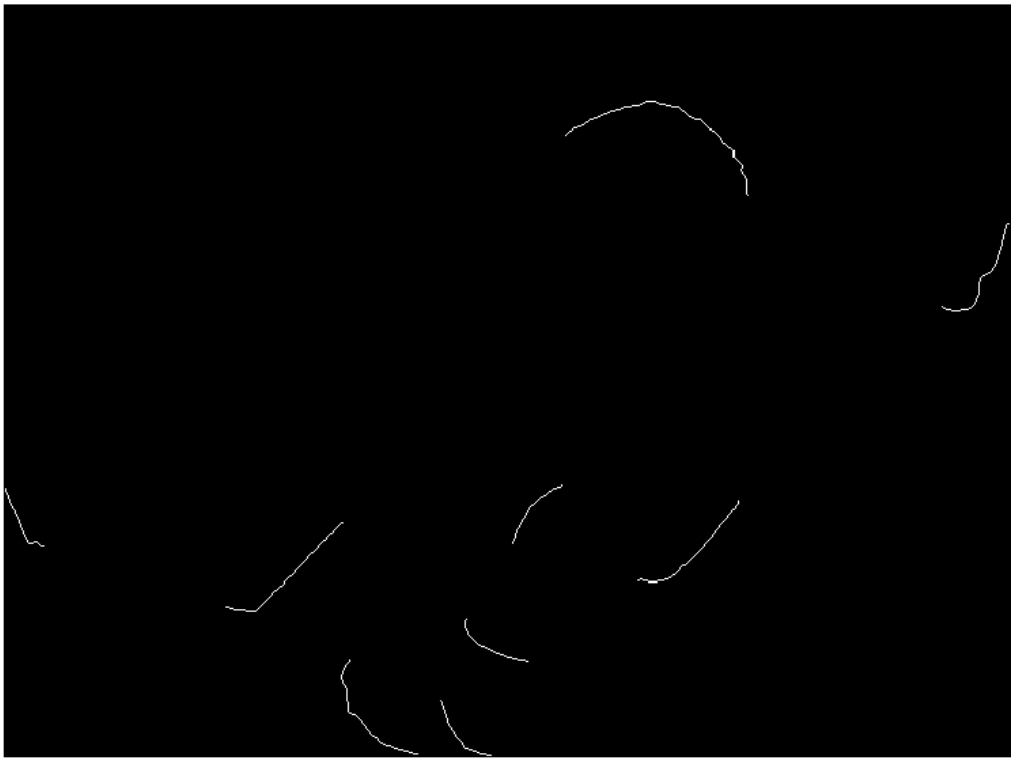


Figure 4.19: Remaining curves obtained after RMS error calculation.

RMS error calculation is as shown in Figure 4.18. RMS error is calculated on only those curves whose radius (R) is found to be more than 40 pixels. The calculated RMS error is shown in the first pixel of each curve. It can be seen from Figure 4.18 that circular curves have less RMS error and thus curves having less RMS error is selected for next stage. After root mean square error

calculation and comparison of radius of different fitted circle with that of average human head radius, Figure 4.19 is obtained.

4.7 Skin color Detection

After using information related to the shape of human head, information related to color is analyzed for human head detection. As human face covers most of human head, color of face can be used to locate human head. Color information can be used to segment objects having one color from the rest. By separating the skin color from the rest, the search area for human face can be limited. The result of segmenting skin colored pixel from the background is as shown in Figure 4.20.

Figure 4.20: Skin color thresholded image.

Although the input image from the camera is in RGB color space, the color space of the input image can be changed to other color spaces. In this thesis, for experimental purpose different

color spaces; L*a*b*, HSV, YUV and YcbCr, was used initially for human skin color detection. The performance of L*a*b*, HSV, YCbCr and YUV in skin color detection is reasonably well, whereas, RGB color space doesn't perform well. It is also reported in [131] that RGB color space is not frequently used for color based object detection and color analysis. Compared to other color spaces, L*a*b* color space was found to perform better in differentiating skin color from the rest and thus is used in this research. The L*a*b* or CIELab color space is an international standard for color measurement introduced in 1976 [132].

For skin color detection, the individual color component such *L*, *a* and *b* in case of Lab color space is checked. If a pixel has all three of its color component within the given range, then the pixel is assumed to be of skin color. If a pixel at some location (10,10) in image plane has the individual color component as *L*=90, *a*=126 and *b*=110. If the threshold for skin color detection is set at *L* (87-95), *a* (125- 130) and *b* (106-116), the pixel is within the threshold and is thus considered as skin colored pixel. Otherwise, it is not considered to be skin colored pixel. The threshold for skin color detection can be determined by investigating skin color of different persons. As skin color is not present only in face, other objects might also have similar color as that of human skin. After finding skin color, it is important to check presence of skin colored pixel on the fitted circle from the previous step. The combination of information about color and shape gives the final location of human head. The verification process after combination of color and shape information is explained in the next step.

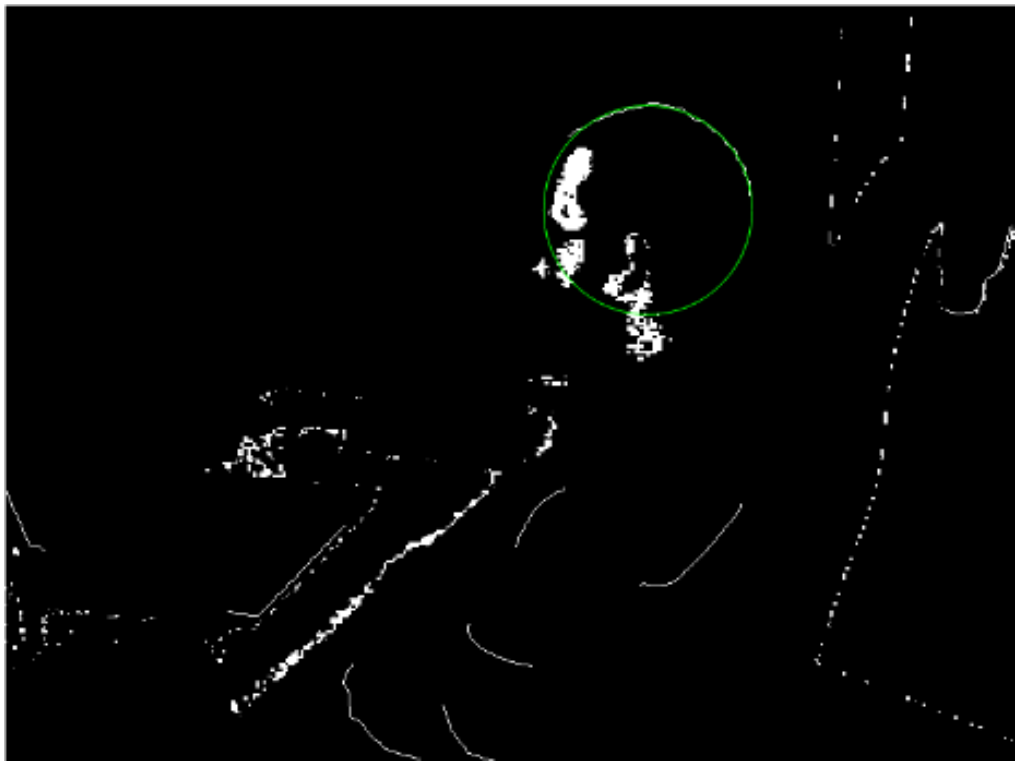


Figure 4.21: Circle fitting on a curve containing skin colored pixel.

4.8 Human Head Verification

Finally, verification process is carried out to see if the curve identified is human head or not. For verification process, presence of skin color is checked on the fitted circle. If the maximum area of the blob is greater than threshold, the blob is considered to represent human face and consequently, the fitted circle encircles the human head. In this research, the threshold is set at 100 pixels, however it may be adjusted according to requirement and the result obtained. The result of final head detection is shown in Figure 4.21. After verification process, human motion can be tracked and its distance from the camera can be calculated using object distance method recently proposed in [133].

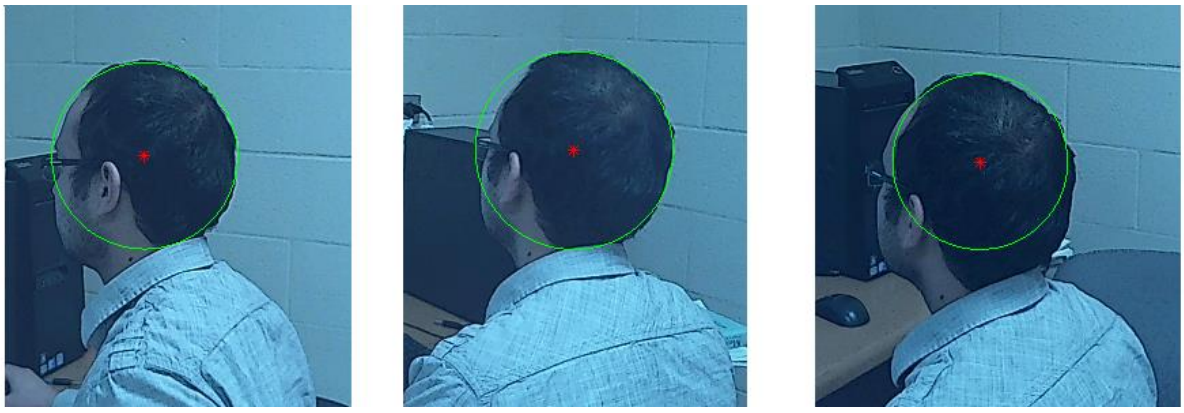


Figure 4.22: Final Head Detection.

The entire steps taken for Human Head detection can be summarized as below:

Step 1: Get array of edges from Edgelink program

Input Image from Camera

Calculate the total number of individual edges in the given image

Output an array of edges

Step 2: From the array of edges,

Calculate two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ on the edge and calculate the angle between them using their slope

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

If the angle difference between consecutive edges is less than threshold delete the segments.

Step 3: Use line fitting as additional criteria to eliminate linear segments which weren't eliminated from step 2.

Step 4: Divide the new edge segments into curves having length of 50 pixels.

Find three points (x_1,y_1), (x_2,y_2) and (x_3,y_3) on those curves and calculate the curvature

$$\kappa = \frac{2\left|(x_2 - x_1).X(y_3 - y_1) - (x_3 - x_1).X(y_2 - y_1)\right|}{\sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)X((x_3 - x_1)^2 + (y_3 - y_1)^2)X((x_3 - x_2)^2 + (y_3 - y_2)^2)}}$$

Step 5: Delete segments whose curvature (k) is not close to that of human head

If abs($\kappa_{des} - \kappa$)> 0.007

Delete the segment

Step 6: Apply circle fitting on the remaining edges, and consider only those curves having radius (R) >40 pixels and RMS error below threshold for further processing

Step 7: Create a binary image by finding skin colored pixel in the image
Input RGB image
Convert RGB color space to L*a*b* color space
If L*, a* and b* values of individual pixel lies within the threshold, it will be classified as skin colored pixel.

Step8: If the fitted circle contains skin colored blob with area greater than threshold, finalize head detection.

Step 9: Start Tracking

4.9 Summary

In this chapter, human detection method using curvature information and skin color was presented. Further, it was shown that curvature criteria could be used to detect human head after applying curvature criteria to detect curvature of circular objects. To increase the accuracy of human detection method, two criteria i.e., shape and color information was used. Different challenges on human detection in indoor environment was presented and possible solutions were also explained. After detection of human, recognition could be the next stage. In the next chapter face recognition method on a small population would be discussed.

Chapter 5

FACE RECOGNITION AND RESULT

5.1 Introduction

Once human is detected by computer vision system linked to the robot, the next step would be to recognize them. Face recognition can be either an identification or a verification problem [34]. Identification problem is to identify a person based on the information on the stored database. Whereas, in verification the identity entered into the method is checked and verified with the help of the information from the database. In Chapter 2, it was explained that face recognition system can be based on the whole image or it can also be based on the face features. A feature based method is followed on this thesis. Face recognition system in general has following steps:

a. Face Detection

b. Feature Point Detection

c. Feature Extraction

d. Classification

In this thesis, for the purpose of training and face recognition a small size of population of eight persons is considered. The experiment carried out in this thesis was to see if choosing fewer number of discriminative key points would be sufficient for face recognition for a small population. To check if choosing a few number of key points would be sufficient for recognition task, eight subjects were chosen as shown in Figure 5.1. in which faces of the subjects have been blurred. For face recognition, face detection is the initial step which is explained next.

Figure 5.1: Subjects in Face Recognition.

5.2 Face Detection

Face Detection is the preliminary step for face recognition system. The task of face detection algorithm is to confirm if there is any face on the image plane and localize them if it is there. As, the Face detection process provides the size and location of the face, the accuracy and reliability of face recognition system also depends on the chosen face detection process. In most of the face recognition algorithm, it is assumed that the face is already detected. Most of the face recognition algorithm use Viola-Jones face detection algorithm [134]. Viola-Jones method is a template based face detection algorithm. Face detection using Viola-Jones method is popular because it is easy to implement and it produces a good result. Face detection method using Viola-Jones method is fast and can be used in system requiring real time detection.

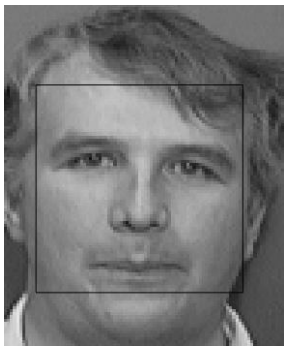


Figure 5.2: Face detection using Viola-Jones Algorithm [134].

Face detection using Viola- Jones algorithm is as shown in Figure 5.2. After face detection, the next step is to locate facial parts such as eyes, mouth, nose and so on and use their location to localize feature points. Although, MATLAB also has detectors to detect eyes, mouth and nose, the performance wasn't found to be satisfactory. Thus, a reliable landmark localization method had to be used before further processing could be carried out. The facial part detection or landmark localization method is described next.

5.3 Landmark Localization

After face detection, localization of facial parts is necessary for analyzing Gabor features at those points. Thus, Landmark localizations is another step necessary before further operations are carried out. Landmark localization or face keypoints localization is very crucial as it determines the success of the face recognition program. Early attempts on landmark localizations was based on gray level information. Although, it could give some good result, the error rate was considerably high even in face database where the images were already pre-processed. In this thesis, the program given in reference [43] is used for landmark localization.

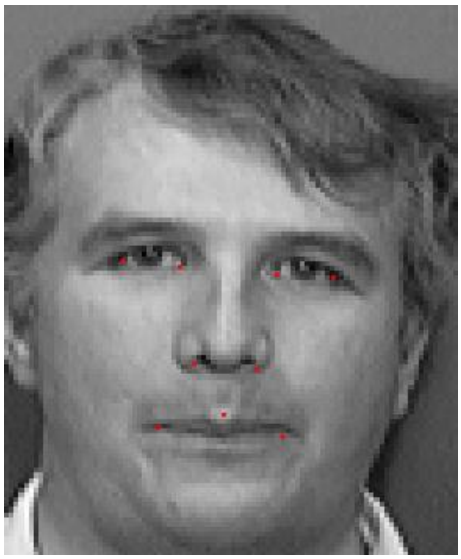


Figure 5.3: Face Keypoints Localization [35].

The performance of this algorithm was checked on the test set [43] with the MATLAB Program. It had a very good localization result and the process was fast. Noticing the good performance of the algorithm, its performance was checked on live image from the camera. For frontal and near frontal view, the program gives accurate result. Figure 5.3 shows successful localization of face keypoints on one of the subjects of AT&T face database. Similarly, Figure 5.4 shows successful localization of keypoints in some of the subjects of this study. In some cases, as in Figure 5.5, the localization isn't perfect. When localization is not perfect, there could be a high chance of wrong classification.

Also, the choice of the fiducial points will affect the result. The corners of the mouth may be covered by moustache and beard sometimes and sometimes it would be shaved thus creating variation. Similarly, corner of the nose would be highly sensitive to the angle where the light source is located. Thus, it is necessary to train the classifier using various pose and lighting conditions.

Figure 5.4: Face Keypoints Localization in input image.

Figure 5.5: Incorrect localization of face keypoints [35].

5.4 Feature Extraction

After the Face Keypoints or landmark points are localized, the next step is to construct a feature vector from those key points. To construct feature vector, Gabor filter were applied on the keypoints. Before applying Gabor filter on the image of the subjects, the recognition rate using Gabor filter on subjects of AT&T database was calculated. Gabor filter gave a good recognition rate i.e., above 80%. However, when using AT&T database, the Gabor filter response was taken at five points instead of nine points. With success of application of Gabor filter on AT&T images, Gabor filter was applied on the input image taken from the webcam. For Feature extraction, Gabor filter of different wavelength [2, 16] and orientation [0^0, 45^o, 90^o and 135^o] is applied to the corner of the eyes, nostrils and three points on mouth. The expectation of this program was that, Gabor filter response from those nine points should be sufficient to recognize

eight persons with reasonable accuracy. After applying Gabor filter, the filter response from the input image at those nine points is used to create a feature vector. Here, as Gabor filter of two wavelengths and four orientations are used thus at each keypoint, a 8×1 feature vector is formed. Total nine points; four points around the eye, two on nose and three on mouth is considered thus the final feature vector has 72×1 dimension. One of the subject from AT&T database is shown in Figure 5.6 and the application of Gabor filter on the input image is as shown in Figure 5.7.

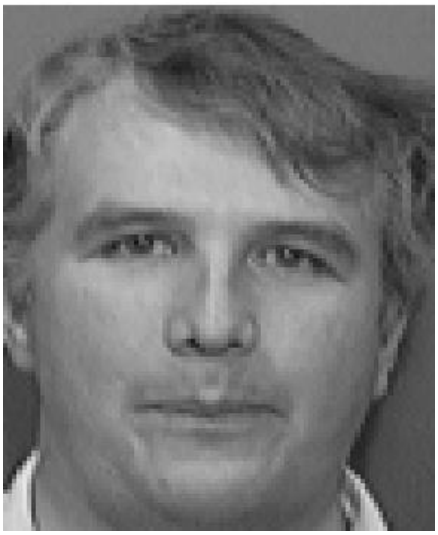


Figure 5.6: Input Image [35].

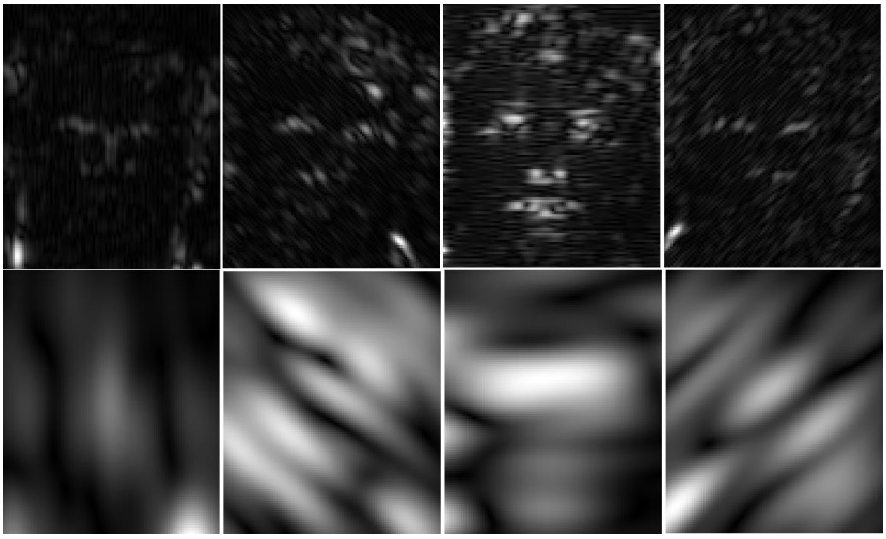


Figure 5.7: Gabor filter response of input image.

In Figure 5.7, the figures in four columns are the Gabor filter responses of four different orientations i.e, [0^0, 45^o, 90^o and 135^o] whereas, two rows represent responses of two different wavelengths i.e., [2, 16]. The feature vectors are then used by a classifier to train a classifier. The classification process is described next.

5.5 Face Classifier

After collecting feature vector of various subjects, the feature could be stored in a database to compare with input image or it could be used to train a classifier. As, the response were taken only from few points, the error rate could be high when comparing the response directly with the data stored in database. Thus, a classifier was trained to classify the input image to one of the eight classes. Training a Classifier was found to produce a promising result. In this thesis, Support Vector Machine was used as a classifier.

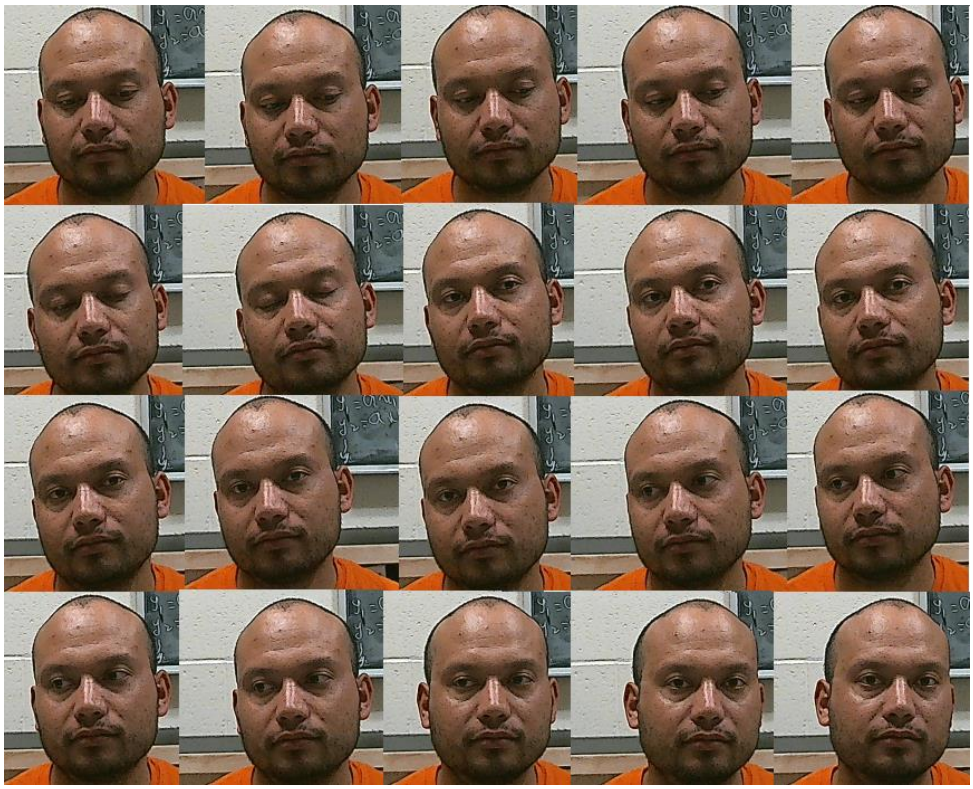


Figure 5.8: Images in a video Sequence.

Figure 5.9: Face Recognition of input image.

5.6 Results

As face Detection and localization of the landmark points are the preliminary step of this method, the performance of the recognition method depends on those methods. A few images in video sequence are shown in Figure 5.8. The recognition rate is calculated based on the frames where face is successfully detected. The recognition result is as shown in Figure 5.9. In some cases, the landmark points were not perfectly localized whereas in some cases it was only partially localized (e.g. only eyes and nose instead of all nine points). As the result of recognition depends upon the texture around the key points, the localization of facial parts has direct result on the recognition process. On image frames where the localization was perfect or near perfect, promising result was obtained. To train the classifier, first 50 frames of the video sequence were

used whereas, the last 100 frames were used for testing. In Table 5.1, the recognition rate is calculated for different numbers of frame as follows:

Table 5.1: Recognition Rate.

Subjects/Frame	30 Frames	50 Frames	100 Frames
Subject 1	100 %	100%	97.96%
Subject2	100%	100%	99%
Subject 3	53.33%	72%	86%
Subject 4	100 %	100 %	100 %
Subject 5	100%	93.88%	89.69%
Subject 6	96.15%	97.8%	98.83%
Subject 7	86.67%	94.12%	97.5%
Subject 8	96.67%	98%	99%

It is seen from the Figure 5.8 that there is some variation in pose, however if there is wide variation in pose then there is error in landmark localization as a result of which the recognition rate suffers. However, if the face landmark localization algorithm can accurately localize the facial parts then the recognition rate will be high. It is worth mentioning that the training images used in this thesis was 50 per subject, it can be assumed that by taking more samples during training phase, the recognition rate can be improved further.

5.7 Summary

In this chapter, a face recognition method on a small database was performed using only the corners of the eye, nose, and mouth. As those points are the most discriminative point in human face, it was found that using those points as feature points gave a satisfactory result (more than

90% in average). The recognition rate calculated on 100 frame sequence suggested that this method can be used in real time. To conclude, it was found in this chapter that for a small database, application of Gabor filter on only few landmark points would be sufficient for face recognition process.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Summary of Thesis and Conclusion

The thesis was divided into two parts. In the first part, human head detection method using single camera was presented and on the second part, face recognition system on a small population using Gabor filter response on nine landmark points was presented.

In the human detection method, it was found that curvature of different objects could be accurately estimated using curvature on different segments of the curve. This could be a new method towards detecting objects having different shapes. After being able to find the curvature of different curves successfully, it was found that human detection could be done with single camera using shape and color information.

On the second part, the use of few points on recognition of different individual was studied. It was found that using only nine points could produce a good recognition rate. However, the recognition rate might suffer when the localization is not perfect. Thus, by including more images on training set if the localization of facial key parts can be made more accurate, the recognition rate will further improve.

6.2 Thesis Contribution

The contribution of this thesis can be divided into two sections. In the first section, human detection method was presented using curvature and color information. Although, curvature property could be used in detection of circles of fixed radius, its implementation to detect digital

curves could be difficult. Some of the challenges were, broken edges of head contour or head edge being linked to other objects in the background. As curvature of Human Head is different in profile and frontal view, the range of reference human head curvature had to include curvature value of human head from both views. This resulted in some other curves being selected as human head contour. Thus, skin color detection was used as secondary criteria for human head detection. To conclude, it was shown in the thesis that curvature criteria along with skin color detection criteria could be used to detect possible human near a robot.

In the second part, the possibility of face recognition on a small database with Gabor filter response on nine face keypoints was investigated. The challenge on face recognition was in some cases the landmark points weren't accurately localized. However, most of the landmark points were correctly localized. The face recognition result for eight subjects in indoor environment suggests that Gabor response from nine points on the human face gives a satisfactory result.

6.3 Recommendation for Future Works

For the future studies, following recommendations could be implemented.

a. Detection of multiple persons and simultaneous face recognition could be performed.
b. The idea of curvature could be used to detect circular or near circular object in manufacturing and production lines.
c. Face recognition using only few points could be experimented for more than eight persons to be utilized in small offices.
d. The result of face recognition can be observed under different lighting conditions.

References:

[1] J. J. Craig, “Introduction to robotics. 1986,” *Read. MA Addison*.

[2] K. Wada, T. Shibata, T. Asada, and T. Musha, “Robot therapy for prevention of dementia at home,” *J. Robot. Mechatron.*, vol. 19, no. 6, pp. 691, 2007.

[3] C. Lin and J. W. Mao-jiun, “Human–robot interaction in an aircraft wing drilling system,” *Int. J. Ind. Ergon.*, vol. 23, no. 1, pp. 83–94, 1999.

[4] R. Bemelmans, G. J. Gelderblom, P. Jonker, and L. De Witte, “Socially assistive robots in elderly care: A systematic review into effects and effectiveness,” *J. Am. Med. Dir. Assoc.*, vol. 13, no. 2, pp. 114–120, 2012.

[5] Y. Sankai, “The Leading Edge of Future Technology‘ Cybernics’: Project HAL-Toward Robot Suits and Cyber Suits?,” in *Ninth IEEE International Symposium on Wearable Computers (ISWC’05)*, 2005, pp. xvii–xvii.

[6] I. Ulrich, F. Mondada, and J.-D. Nicoud, “Autonomous vacuum cleaner,” *Robot. Auton. Syst.*, vol. 19, no. 3, pp. 233–245, 1997.

[7] P. J. Hinds, T. L. Roberts, and H. Jones, “Whose job is it anyway? A study of human-robot interaction in a collaborative task,” *Hum.-Comput. Interact.*, vol. 19, no. 1, pp. 151–181, 2004.

[8] M. Rahimi and W. Karwowski, “A research paradigm in human-robot interaction,” *Int. J. Ind. Ergon.*, vol. 5, no. 1, pp. 59–71, 1990.

[9] D. Y. Y. Sim and C. K. Loo, “Extensive assessment and evaluation methodologies on assistive social robots for modelling human–robot interaction–A review,” *Inf. Sci.*, vol. 301, pp. 305–344, 2015.

[10] M. A. Goodrich and A. C. Schultz, “Human-robot interaction: a survey,” *Found. Trends Hum.-Comput. Interact.*, vol. 1, no. 3, pp. 203–275, 2007.

[11] M. Salem, F. Eyssel, K. Rohlfing, S. Kopp, and F. Joublin, “To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability,” *Int. J. Soc. Robot.*, vol. 5, no. 3, pp. 313–323, 2013.

[12] R. Pfeifer, F. Iida, and M. Lungarella, “Cognition from the bottom up: on biological inspiration, body morphology, and soft materials,” *Trends Cogn. Sci.*, vol. 18, no. 8, pp. 404–413, 2014.

[13] R. Pfeifer, M. Lungarella, and F. Iida, “The challenges ahead for bio-inspired’soft’robotics,” *Commun. ACM*, vol. 55, no. 11, pp. 76–87, 2012.

[14] T. Spexard, A. Haasch, J. Fritsch, and G. Sagerer, “Human-like person tracking with an anthropomorphic robot,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 1286–1292.

[15] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, “Collision detection and reaction: A contribution to safe physical human-robot interaction,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3356–3363.

[16] H.-J. Böhme *et al.*, “An approach to multi-modal human–machine interaction for intelligent service robots,” *Robot. Auton. Syst.*, vol. 44, no. 1, pp. 83–96, 2003.

[17] L. Spinello and R. Siegwart, “Human detection using multimodal and multidimensional features,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 3264–3269.

[18] P. H. Torr and A. Zisserman, “Latent svms for human detection with a locally affine deformation field,” 2012.

[19] N. A. Ogale, "A survey of techniques for human detection from video," *Surv. Univ. Md.*, vol. 125, p. 133, 2006.

[20] M. P. Parmar and M. G. S. Pandi, "Performance Analysis and Augmentation of K-means Clustering, based approach for Human Detection in Videos," in *International Journal of Engineering Development and Research*, 2015, vol. 3.

[21] J. Zhou and J. Hoang, "Real time robust human detection and tracking system," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, 2005, pp. 149–149.

[22] J. Ruiz-del-Solar and R. Verschae, "Robust skin segmentation using neighborhood information," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, 2004, vol. 1, pp. 207–210.

[23] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, 1998, pp. 232–237.

[24] S. Nigam and A. Khare, "Multiresolution approach for multiple human detection using moments and local binary patterns," *Multimed. Tools Appl.*, vol. 74, no. 17, pp. 7037–7062, 2015.

[25] J. Yang, W. Lu, and A. Waibel, "Skin-color modeling and adaptation," in *Asian Conference on Computer Vision*, 1998, pp. 687–694.

[26] E. Martinson and V. Yalla, "Augmenting deep convolutional neural networks with depth-based layered detection for human detection," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016, pp. 1073–1078.

[27] N. Bellotto and H. Hu, “Multisensor-based human detection and tracking for mobile service robots,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 39, no. 1, pp. 167–181, 2009.

[28] G. Cielniak and T. Duckett, “Person identification by mobile robots in indoor environments,” in *Robotic Sensing, 2003. ROSE’03. 1st International Workshop on*, Sweden, 2003, .

[29] M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del-Solar, “Human detection and identification by robots using thermal and visual information in domestic environments,” *J. Intell. Robot. Syst.*, vol. 66, no. 1–2, pp. 223–243, 2012.

[30] S. S. Ghidary, Y. Nakata, T. Takamori, and M. Hattori, “Human detection and localization at indoor environment by home robot,” in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, 2000, vol. 2, pp. 1360–1365.

[31] T. Zhang, Z. Yang, W. Jia, Q. Wu, J. Yang, and X. He, “Fast and robust head detection with arbitrary pose and occlusion,” *Multimed. Tools Appl.*, vol. 74, no. 21, pp. 9365–9385, 2015.

[32] W. Zou, Y. Li, K. Yuan, and D. Xu, “Real-time elliptical head contour detection under arbitrary pose and wide distance range,” *J. Vis. Commun. Image Represent.*, vol. 20, no. 3, pp. 217–228, 2009.

[33] W. ZHAO, R. CHELLAPPA, P. PHILLIPS, and A. ROSENFELD, “Face Recognition: A Literature Survey,” *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.

[34] R. Jafri and H. R. Arabnia, “A survey of face recognition techniques.,” 2009.

[35] AT&T, L. Cambridge, “The database of faces,” *Camb. Univ. Comput. Lab. Available, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.*

[36] T. Kanade, “Picture processing system by computer complex and recognition of human faces,” *Dr. Diss. Kyoto Univ.*, vol. 3952, pp. 83–97, 1973.

[37] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[38] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models-their training and application,” *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[39] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, “Feature extraction from faces using deformable templates,” *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 99–111, 1992.

[40] T. F. Cootes, G. J. Edwards, C. J. Taylor, and others, “Active appearance models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[41] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von Der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, 1997.

[42] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 2879–2886.

[43] G. Tzimiropoulos and M. Pantic, “Optimization problems for fast aam fitting in-the-wild,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 593–600.

[44] L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces,” *Josa A*, vol. 4, no. 3, pp. 519–524, 1987.

[45] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR’91., IEEE Computer Society Conference on*, 1991, pp. 586–591.

[46] S. Akamatsu, T. Sasaki, H. Fukamachi, and Y. Suenaga, “Robust face identification scheme: KL expansion of an invariant feature space,” in *Intelligent Robots and Computer Vision X: Algorithms and Techniques*, 1992, pp. 71–84.

[47] A. Lemieux and M. Parizeau, “Experiments on eigenfaces robustness,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2002, vol. 1, pp. 421–424.

[48] A. Pentland, B. Moghaddam, and T. Starner, “View-based and modular eigenspaces for face recognition,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*, 1994, pp. 84–91.

[49] M.-H. Yang, N. Ahuja, and D. Kriegman, “Face recognition using kernel eigenfaces,” in *Image processing, 2000. proceedings. 2000 international conference on*, 2000, vol. 1, pp. 37–40.

[50] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[51] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, “A new LDA-based face recognition system which can solve the small sample size problem,” *Pattern Recognit.*, vol. 33, no. 10, pp. 1713–1726, 2000.

[52] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, 2002.

[53] R. Lotlikar and R. Kothari, "Fractional-step dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 623–627, 2000.

[54] J. Yang, A. F. Frangi, J. Yang, D. Zhang, and Z. Jin, "KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 230–244, 2005.

[55] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Advances in neural information processing systems*, 2004, pp. 1569–1576.

[56] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1450–1464, 2002.

[57] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, 2005.

[58] A. Eleyan and H. Demirel, "Face recognition system based on PCA and feedforward neural networks," in *International Work-Conference on Artificial Neural Networks*, 2005, pp. 935–942.

[59] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 697–710, 2002.

[60] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognit.*, vol. 35, no. 10, pp. 2279–2301, 2002.

[61] G. Hu *et al.*, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 142–150.

[62] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, 1997.

[63] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015, vol. 1, p. 6.

[64] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *ArXiv Prepr. ArXiv150200873*, 2015.

[65] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.

[66] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[67] E. Zhou, Z. Cao, and Q. Yin, "Naive-deep face recognition: Touching the limit of LFW benchmark or not?," *ArXiv Prepr. ArXiv150104690*, 2015.

[68] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Web-scale training for face identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2746–2754.

[69] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen, "Image sets alignment for video-based face recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 2626–2633.

[70] M. Yang, P. Zhu, L. Van Gool, and L. Zhang, “Face recognition based on regularized nearest points between image sets,” in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, 2013, pp. 1–7.

[71] P. Zhu, W. Zuo, L. Zhang, S. C.-K. Shiu, and D. Zhang, “Image set-based collaborative representation for face recognition,” *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 7, pp. 1120–1132, 2014.

[72] H. Cevikalp and B. Triggs, “Face recognition based on image sets,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2567–2573.

[73] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, “Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2048–2057.

[74] D. Gong, K. Zhu, Z. Li, and Y. Qiao, “A semantic model for video based face recognition,” in *Information and Automation (ICIA), 2013 IEEE International Conference on*, 2013, pp. 1369–1374.

[75] M. Everingham, J. Sivic, and A. Zisserman, “Taking the bite out of automated naming of characters in TV video,” *Image Vis. Comput.*, vol. 27, no. 5, pp. 545–559, 2009.

[76] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, “Video-based face recognition using probabilistic appearance manifolds,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, vol. 1, p. I–313.

[77] O. Arandjelović and R. Cipolla, “A pose-wise linear illumination manifold model for face recognition using video,” *Comput. Vis. Image Underst.*, vol. 113, no. 1, pp. 113–125, 2009.

[78] A. Monroy, A. Eigenstetter, and B. Ommer, “Beyond straight lines—object detection using curvature,” in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 3561–3564.

[79] H. B. Phillips, *Analytic geometry and calculus*. Addison-Wesley Press, 1942.

[80] M. Worring and A. W. Smeulders, “Digital curvature estimation,” *CVGIP Image Underst.*, vol. 58, no. 3, pp. 366–382, 1993.

[81] A. Belyaev, “Plane and space curves. curvature. curvature–based features,” *Max–Planck–Institut Für Inform.*, 2004.

[82] S. Arca, P. Campadelli, and R. Lanzarotti, “A face recognition system based on local feature analysis,” in *International Conference on Audio-and Video-based Biometric Person Authentication*, 2003, pp. 182–189.

[83] A. S. M. Sohail and P. Bhattacharya, “Detection of facial feature points using anthropometric face model,” in *Signal Processing for Image Enhancement and Multimedia Processing*, Springer, 2008, pp. 189–200.

[84] A. Yilmaz and M. A. Shah, “Automatic feature detection and pose recovery for faces,” in *The 5th Asian Conference on Computer Vision*, 2002, vol. 289.

[85] J. B. Waite and W. J. Welsh, “An application of active contour models to head boundary location.,” in *BMVC*, 1990, pp. 1–6.

[86] I. Craw, D. Tock, and A. Bennett, “Finding face features,” in *European Conference on Computer Vision*, 1992, pp. 92–96.

[87] R. Radke, *Active Shape Models*. 2015.

[88] O. Çeliktutan, S. Ulukaya, and B. Sankur, “A comparative study of face landmarking techniques,” *EURASIP J. Image Video Process.*, vol. 2013, no. 1, p. 1, 2013.

[89] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 1003–1011.

[90] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 12, pp. 2091–2110, 1989.

[91] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, 1962.

[92] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 7, pp. 1169–1179, 1988.

[93] M. Li and R. C. Staunton, "Optimum Gabor filter design and local binary patterns for texture segmentation," *Pattern Recognit. Lett.*, vol. 29, no. 5, pp. 664–672, 2008.

[94] M. Zhou and H. Wei, "Face verification using gaborwavelets and adaboost," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, vol. 1, pp. 404–407.

[95] E. Hjelmas and J. Wroldsen, "Recognizing faces from the eyes only," in *In Proceedings of the 11th Scandinavian Conference on Image Analysis*, 1999.

[96] L. Shen and L. Bai, "A review on Gabor wavelets for face recognition," *Pattern Anal. Appl.*, vol. 9, no. 2–3, pp. 273–292, 2006.

[97] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachida, "Face and facial feature extraction from color image," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, 1996, pp. 345–350.

[98] D. J. Beymer, “Face recognition under varying pose,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*, 1994, pp. 756–761.

[99] X. Lu and A. K. Jain, “Multimodal facial feature extraction for automatic 3D face recognition,” *Dep. Comput. Sci. Mich. State Univ. East Lansing Mich. Tech Rep MSU-CSE-05-22*, 2005.

[100] D. A. Clausi and M. E. Jernigan, “Designing Gabor filters for optimal texture separability,” *Pattern Recognit.*, vol. 33, no. 11, pp. 1835–1849, 2000.

[101] A.-A. Bhuiyan and C. H. Liu, “On face recognition using gabor filters,” *World Acad. Sci. Eng. Technol.*, vol. 28, pp. 51–56, 2007.

[102] W. Zhang, S. Shan, L. Qing, X. Chen, and W. Gao, “Are Gabor phases really useless for face recognition?,” *Pattern Anal. Appl.*, vol. 12, no. 3, pp. 301–307, 2009.

[103] S. F. Hafez, M. M. Selim, and H. H. Zayed, “2D Face Recognition System Based on Selected Gabor Filters and Linear Discriminant Analysis LDA,” *ArXiv Prepr. ArXiv150303741*, 2015.

[104] S. Xie, S. Shan, X. Chen, and J. Chen, “Fusing local patterns of gabor magnitude and phase for face recognition,” *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1349–1361, 2010.

[105] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, A. Smola, and others, “Support vector machine-reference manual,” 1998.

[106] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

[107] L. Wang, *Support vector machines: theory and applications*, vol. 177. Springer Science & Business Media, 2005.

[108] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[109] C. Huang, L. S. Davis, and J. R. G. Townshend, "An assessment of support vector machines for land cover classification," *Int. J. Remote Sens.*, vol. 23, no. 4, pp. 725–749, 2002.

[110] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[111] R. Berwick, "An Idiot's guide to Support vector machines (SVMs)," *Retrieved Oct.*, vol. 21, p. 2011, 2003.

[112] X. Bai *et al.*, "Learning ECOC code matrix for multiclass classification with application to glaucoma diagnosis," *J. Med. Syst.*, vol. 40, no. 4, pp. 1–10, 2016.

[113] K.-B. Duan and S. S. Keerthi, "Which is the best multiclass SVM method? An empirical study," in *International Workshop on Multiple Classifier Systems*, 2005, pp. 278–285.

[114] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 285–297, 2009.

[115] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 120–134, 2010.

[116] M. J. Escobar and J. Ruiz-del-Solar, "Biologically-based face recognition using Gabor filters and log-polar images," in *Proceedings of the International Joint Conference on Neural Networks*, 2002, vol. 2, pp. 1143–1147.

[117] Y. B. Jemaa and S. Khanfir, "Automatic local Gabor features extraction for face recognition," *ArXiv Prepr. ArXiv09074984*, 2009.

[118] P. J. Phillips and P. Rauss, "The face recognition technology (FERET) program," in *CTAC International Technology Symposium, Chicago, Illinois, August*, 1997, pp. 18–22.

[119] X. Tan and B. Triggs, "Fusing Gabor and LBP feature sets for kernel-based face recognition," in *International Workshop on Analysis and Modeling of Faces and Gestures*, 2007, pp. 235–249.

[120] E. R. Davies, *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.

[121] M. A. Oskoei and H. Hu, "A survey on edge detection methods," *Univ. Essex Essex UK*, 2010.

[122] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 679–698, 1986.

[123] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1485–1490, 2005.

[124] B. Green, "Canny edge detection tutorial," *DOI Httpwww Pages Drexel Edu∼ Weg22cantut Html*, 2002.

[125] L. A. Fernandes and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," *Pattern Recognit.*, vol. 41, no. 1, pp. 299–314, 2008.

[126] L. Liu, D. Zhang, and J. You, "Detecting wide lines using isotropic nonlinear filtering," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1584–1595, 2007.

[127] P. Kovesi, *MATLAB and Octave Functions for Computer Vision and Image Processing*. 2000.

[128] M. F. Catapan, M. L. L. Okimoto, F. E. Santana, C. M. Silva, Y. W. Rodrigues, and others, “Anthropometric Analysis of Human Head for Designing Ballistic Helmets,” *Procedia Manuf.*, vol. 3, pp. 5475–5481, 2015.

[129] M. F. Catapan, “Análise antropométrica da cabeça humana para dimensionamento de capacetes balísticos,” 2014.

[130] N. Chernov and C. Lesort, “Least squares fitting of circles and lines,” *ArXiv Prepr. Cs0301001*, 2003.

[131] K. B. Shaik, P. Ganesan, V. Kalist, B. S. Sathish, and J. M. M. Jenitha, “Comparative study of skin color detection and segmentation in HSV and YCbCr color space,” *Procedia Comput. Sci.*, vol. 57, pp. 41–48, 2015.

[132] K. Leon, D. Mery, F. Pedreschi, and J. Leon, “Color measurement in L∗ a∗ b∗ units from RGB digital images,” *Food Res. Int.*, vol. 39, no. 10, pp. 1084–1091, 2006.

[133] P. Alizadeh and M. Zeinali, “A Real-Time Object Distance Measurement Using a Monocular Camera" Proceedings of 24th IASTED international Conference on Modelling and Simulation, Canada, Alberta, Banff, July 17, 2013.

[134] P. Viola and M. J. Jones, “Robust real-time face detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.